Archetype based intelligent system
for healthcare interoperability

Luís Miguel Mesquita Miranda

UMinho | 2012

Universidade do Minho
Escola de Engenharia

Luís Miguel Mesquita Miranda

Archetype based intelligent system for healthcare interoperability

Outubro de 2012

Universidade do Minho
Escola de Engenharia

Luís Miguel Mesquita Miranda

Archetype based intelligent system for healthcare interoperability

Programa Doutoral em Engenharia Biomédica

Trabalho realizado sob a orientação do
Professor Doutor José Carlos Ferreira Maia Neves
e do
Professor Doutor José Manuel Ferreira Machado

Outubro de 2012

DECLARAÇÃO

Nome: Luís Miguel Mesquita Miranda

Correio electrónico: miranda@di.uminho.pt

Telefone: 960 387 449

Número do Cartão de Cidadão: 12915226

Título da tese de doutoramento:
Archetype based intelligent system for healthcare interoperability

Orientadores:

Professor Doutor José Carlos Ferreira Maia Neves

Professor Doutor José Manuel Ferreira Machado

Designação do Ramo de Conhecimento do Doutoramento:

Programa Doutoral em Engenharia Biomédica

Universidade do Minho, 20 de Outubro de 2012

Assinatura: ____

Acknowledgements

My growth during this research was filled with challenges and difficulties I could not have overcome without the help of family and friends. Hopping not to let anyone forgotten, these paragraphs will be dedicated to those special persons that I really need to express my gratitude.

Firstly I want to address my words go to my two supervisors, for all their patience and dedication to my work and problems, Professor José Carlos Maia Neves and Professor José Machado. Without their dedication to our work together and advices towards my difficulties, this research would not be possible. I also need to mention Professor António Abelha and Manuel Filipe Santos for all their dedication during my University formation and work at the numerous hospital centers we worked on.

From the bottom of my heart I want to thank my family, namely my mother, father and brother. They incited me to fight hard for my goals, giving me always support to achieve them.

I have good and unique friends that are a major piece of my life and without whom I'd not be the same person. They inspired me and held me together in the harsh moments that are a always part of a PhD. Firstly I want to address Olia Goda. We've known each other since our teens, and since then in both good and bad moments, I can always count in you Olia, to put me in perspective and share with me your strength when I lack mine. True friends never go away and you are a strong part of my resolve. Having you in my life was one of the best things fate brought me. Also my cradle parter João Afonso. João, since our childhood we've known and picked on each other as brothers. From child's prank to adult's problem, thanks for being there. Thirdly, I must thank my dear friend Andreea Urzica for shedding some of her enlightenment into this otherwise lost soul. Moreover, I must mention my close friends Ana Pontes, Cláudia Martins, Ângelo Costa, Júlio Duarte, Zé Leite, Paulo Pinto, Manuel Nascimento, Jean-Pierre and Nillufer Çokak. You were essential to me and I hope will continue to be.

To the members of the *Centro Hospitalar do Porto*, *Centro Hospitalar do Tâmega e Sousa*, *Centro Hopitalar do Alto Ave* and *Unidade Local de Saúde do Norte Alentejano*, that aided me in this research project, my everlasting gratitude. I learnt a lot from this experience.

This PhD project is supported by the grant **SFRH/BD/65023/2009** of the Portuguese Foundation for Science and Technology (FCT)

Abstract

The healthcare arena configures an environment of both complexity and cooperation, in which numerous and distinct information systems must exchange information in a expedite and consolidated manner. Where healthcare interoperability is concerned several techniques, methodologies, architectures and standards exist. However subjects such as service distribution, fault tolerance, standards, communication *flavoring* and tightly-bound systems still are a major issue of concern.

This work studies and researches the best methodologies to imbue intelligent behaviours combined with ontology and moral awareness into multi-agents system applied to healthcare environments. Its core objective is to propose, develop, implement and evaluate an archetype for an interoperability platform oriented towards the healthcare environment. This archetype was validated in several implementation in different major healthcare institutions. It is based in an agent framework named JADE and is adapted and oriented towards the healthcare environment.

Henceforth the resulting archetype addresses the existing limitations in past and present solutions regarding healthcare interoperability. It explores the limits of intelligent behaviours in multi-agent systems applied to interoperation procedures in healthcare, towards the improvement of the reliability and quality of information exchanged.

Sumário

A área da saúde configura um ambiente de grande complexidade e cooperação onde inúmeros e distintos sistemas de informação têm que trocar informação entre si de uma forma expedita e consolidada. No âmbito da interoperabilidade hospitalar existem várias técnicas, metodologias, arquiteturas e standards. No entanto, temas como distribuição de serviços, tolerância à falha, standards, *flavouring* de comunicações e sistemas fortemente acoplados, continuam a ser um importante fonte de preocupação.

Este trabalho estuda e pesquiza as melhores metodologias de embeber comportamentos inteligentes combinados com ontologias e noções morais em sistemas multi-agentes aplicados a ambientes hospitalares. O seu objectivo principal é propor, desenvolver, implementar e avaliar um arquétipo para uma plataforma de interoperabilidade orientada para o ambiente hospitalar. Este arquétipo foi validado em diferentes implementações em instituições de saúde portuguesas de grande dimensão. Esta plataforma é baseada numa framework de agentes denominada JADE e foi adaptada e orientada para o ambiente hospitalar..

Desta forma o arquétipo resultante é orientado para resolver as limitações existentes nas soluções atuais de interoperabilidade hospitalar. Este explora os limites dos comportamentos inteligentes em sistemas multi-agente quando aplicados em procedimentos de interoperabilidade na área da saúde para melhorar a fiabilidade e qualidade da informação trocada entre estes sistemas.

Contents

Acronyms

ACL Agent Communication Language

AI Artificial Intelligence

AIDA *Agência para Interoperabilididade Difusão e Arquivo*

AMS Agent Management System

CICA *Centro de Cirurgia de Ambulatório*

CHP *Centro Hospitalar do Porto*

DCMI Dublin Core Metadata Initiative

DF Directory Facilitator

DSS Decision Support Systems

EHR Electronic Health Record

HIS Health Information Systems

HL7 Health Level Seven

LIS Laboratory Information System

MAS Multi-Agent System

PMS Patient Management System

RIS Radiological Information System

SL Semantic Language

SOA Service Oriented Architecture

WHO World Health Organization

XML Extensible Markup Language

XSD XML Schema Definition

List of Figures

List of Tables

Chapter 1

Introduction

When considering the core objectives of technology and engineering, the aim to solve existing problems and improve otherwise limited solutions, emerges as one of the most valuable ones. Such is the value of information technologies in healthcare, as procedures and patient care are improved through its usage. Currently, information technologies acquired a key role on the flow of work, information and knowledge within healthcare institutions determining their inner functioning to an extent previously unexpected. The uniqueness of each service provided and technology implemented require particularly adapted Information Systems (IS), which need to exchange information and guarantee both information quality and effective technical capabilities, such as standardisation, modularity, extensibility, distributability and interoperability. (Bodart et al, 2000) Furthermore, the consolidation of patient information and knowledge is an increasingly tool to improve the quality of service; to reduce costs on the long run and an essential pre-requisite for the development of group decision support systems. (Miranda et al, 2009a)

The dissemination of information technologies in the healthcare arena has been increasingly visible in the daily practices of healthcare institutions, mainly those related to patient clinical record. This tendency resulted in several solutions for providing support in the regular decisions clinical staff is presented with .(Dupuits and Hasman, 1995) (Dreiseitl and Binder, 2005) According to (Dreiseitl and Binder, 2005), the benefits expectable from the association of such technologies are related with :

- the improvement of the quality of the treatment providing automated warnings and and verification of information consistency;
- an increase of the effectiveness by providing better guidelines for clinical registry;
- an increase in the knowledge available and possibility to provide the right information wanted when and where is is necessary; and
- reduction of costs on the long run, regarding costs of the information management and realisation of complementary diagnose methods, removing the necessity for redundant exams.

The overall benefits of information systems in healthcare are however undermined by the complexity of making each of these systems communicate in a loosely bounded manner. From the diversity of existing solutions, each of them oriented towards a service or group of services, scattered information that is vital to be shared is often secluded or connected in an intricate bound.

The concepts of interoperability and integration are nowadays a cornerstone to information systems in healthcare. As it is more thoroughly explained in Chapter 2, although both work based on entangled synergy, interoperability and integration address complementary issues using distinct technologies and methodologies.

In the specific case of healthcare interoperability and integration, not only technological advances have occurred. Several terminologies and ontologies, that can be embedded in software applications for the most distinct reasons, have been developed by international committees. The area of medical informatics is perhaps one the most standardised and rich in the area of interoperability and integration. Furthermore, several solutions and research projects presented in the following sections use artificial intelligence paradigms, such as the agent oriented paradigm to modularise and implement an interoperability in healthcare.

The existence of these resources enables the possibility to use agent argumentation protocols and paradigms associated with ontologies towards semantic reasoning and to further enhance the interoperability and integrations processes within healthcare. Although some technological advances such as HL7 already aim to introduce semantic and conceptual notion of interoperability, their implementation is far from being disseminated and their relational models are not a pure and unequivocal reflection of the information exchanged in healthcare.

The following thesis presents a line of research that aims to use intelligent systems in order to overcome some of the previously mentioned limitation that are shed into light with current research and needs found directly in the field. Henceforth, it aims to propose and validate methodologies and develop a prototype aimed towards the introduction of intelligent behaviours in healthcare multi-agent systems.

Although this is an independent research thesis, this work comes in line and integrated in the AIDA platform, and integration platform developed by a small team of researchers of the University of Minho. This platform is in production is in the *Centro Hospitalar do Porto*, *Centro Hospitalar do Alto Ave*, *Centro Hospitalar do Tâmega* e Sousa and *Unidade Local de Saúde do Norte Alentejano*. It was this open door into the existing difficulties in the field that allowed to discover the described research opportunities and to implement the propose archetype and other deliverables in production environment. This represents a great asset to this research and validation of the proposed model.

1.1 Research Opportunities and Objectives

The organisational and functional schema of a healthcare unit configure a distributed computational environment, where different services and people need to be in constant contact and communication, exchanging significant data and knowledge for the purpose of the service they provide (e.g., surgical experiences, patient records, management indicators). Despite their need to communicate, the disparities in their requirements and procedures result in a multitude of different applications, each oriented towards a specific area. Moreover, with the continuous introduction of new medical and information technologies, new systems are being developed in a strong pace.

The resulting growth in information systems transforms the communication among these secluded systems, into a rather complex problem of interoperability. Furthermore, the use of point to point interoperation in large scale healthcare institutions, connecting the immiscible micro-environments within all departments, results in a rather complex web of interconnected communication channels which are difficult to monitor and hardly scalable (Carr and Moore, 2003). These entanglements are a burden on the interoperation among disparate departments of a same institution, but it becomes even more relevant when one moves to the process of interoperation among different institutions, where Business to Business integration (B2B) becomes essential to its proper functioning (i.e. secure sharing of the patients clinical information among different platforms). This is the case of an implementation case described further in the thesis, where within the same legal institutions two entities communicate in an integrate manner.

Indeed, the problem of interoperability is increasingly becoming a cause for major concern within the healthcare environment. The number of healthcare units being in use and the growth in the number of patients in need of care is increasing dramatically. Subsequently the amount of data and information regarding the patient also increases in such a way interoperation without fails and over redundancy becomes a challenge. On the other hand, independently on the conditions under which a problem may occur, a dynamic and pre-emptive response to the disaster or error is needed by the interoperation platform. In other words the system must be capable to detect an abnormal behaviour and predict possible fail, which might be happening, but not have been detected or even failures about to be happening.

Therefore, the research presented in this thesis results from the challenges of medical system interoperation, proposing system integration and communication in a smart environment, in which workflows and communications are dynamic and monitored, in terms of the Health Information System and the Heath Medical Record (HMR).

The complexity of healthcare interoperation also configures an agglomeration of intertwined critical systems, which must be available constantly and must not demonstrate any kind of abnormal behaviour. Monitoring of this characteristics and detection of failure before user notice is hardly attainable regarding the intricacy of the interoperation procedures, opening an opportunity for research in artificial intelligence prediction algorithms for dynamically modelling system configuration rules and settings from agent feed-back, allowing the system to learn how the environment changes, predict critical failures or abnormalities and loss of information.

Henceforth, the another of the basilar aims of this research is to study difficulties in healthcare systems to be able to monitor the existing agents according to their performance, behaviour and life cycle, as well as predicting future complications and taking defined measures to predict minor to critical anomalies. This behaviour that is proposed to achieve represents an important step for intelligent agent life cycles when interoperating distinct multi-agent platforms using the same framework.

The use of terminologies such as SNOMED and ontologies such as UMLS has the potential to enable the implementation of different methodologies towards medical research and medical decision support systems. Recent investigation on ontology based argumentation and logic reasoning through ontology based knowledge base indicates potential for decision support systems and for the quality of information, therefore, an added value for the quality of the decision itself regardless of the objective (clinical, administrative, healthcare policies, ...) (Williams and Hunter, 2007) (Marreiros et al, 2007).

The study for embedding intelligent behaviours into virtual entities has also important synergies with the study of moral reasoning and moral agency. On previous research in (Machado et al, 2009b) and (Machado et al, 2009a), a first exploratory attempt to model moral reasoning within logic programming based agents is theorised. Henceforth a cautious step would be the evaluation of such moral reasoning within a more complex agent framework, which could be easily allow the test deployment of such agents in the healthcare environment. Although not directly intertwined with the main subject of research, such line of study can be of use to parallel with current research in the area of machine ethics.

1.2 Objectives

In the following items the previous contextualized description of objectives is structured in a more linear way, which expresses the research process and path followed.

Although the deliverables of this thesis are diverse and practical, its core objective is to **propose, develop, implement and evaluate an archetype for an interoperability platform oriented towards the healthcare environment**. In another words the development and implementation on different major healthcare institutions of an agent based framework oriented towards the healthcare environment, which embeds in different technologies (.Net Framework, Java, ...) and follows the architecture for multi-agent systems defined by the Foundation for Intelligent Physical Agents (FIPA). Henceforth this archetype addresses the existing limitations in past and present solutions.

The most clear deliverables, are related to each of the phases of the fulfilment of the core objective are going to be now described in more detail. The following phases of the research process describe inner objectives and contain, thought otherwise distinct, intertwined and dependent deliverables.

Interoperability archetype oriented towards healthcare

First of all an analysis of the state of the art when it comes to interoperability and healthcare information systems is of the essence in order to consolidate the problems this model must solve and which opportunities it can take advantage of.

As detailed ahead, the selected model is based in multi-agent systems. The proposed archetype explores the limits and introduces the application of distinct technologies to agent and their communication language associated to existing medical interoperation standards, towards greater levels of interoperability.

Although multi-agent systems are already fairly and successfully used in healthcare interoperability, the level of interoperation and intelligence in usual production frameworks and packaged solutions is still rather low. Communications between existing agents or systems are considerably flavoured according to the communication context, even when following standards such as HL7. The recent advances in medical terminologies and general ontology tools aimed for multi-agent systems enables an easier introduction of semantic interoperability, which is one of the direst needs for an high interoperability level.

When defining a communication standard for a specific interoperation problem in healthcare, HL7 is the most disseminated. As mentioned in Section 2.2.3 the version 3 of this standard aims, among many other objectives, to advance the limits of semantic capabilities in interoperation procedures using its RIM to relate each object and instance. Although no actual implementation of this version of this version has, to my current knowledge, been performed in Portugal and the main usable version remains as the 2.x, the possibilities for semantic interoperation are obvious though its association to agent language. Even without the version 3 the previous version can be modelled and encapsulated in an agent ontology in order to achieve such objectives.

Another line research opportunity arrives from the recurring loss of communication between socket based HL7 communication. Current main distributions of HL7 are based on constantly connected socket ports, through which messages and acknowledgements are sent between client and server. On the existing HL7 architecture part of the AIDA platform (see Chapter 4) it exists sufficient data and know-how to study and theorise and implement intelligent behaviours on these systems to allow them overcome existing limitations, such as loss of communication or decontextualized information. In fact, the loss of communication until the end user notices delays on the system is a concerning problem that is aimed to be tackled with these methodologies.

Following the agent modelling line and improving the proposed architecture on (Miranda et al, 2009b) with the usage and evaluation of different algorithms such as K-Means, the HL7 agents can learn to detect anomalous time frames between messages and interact upon the system to correct such problem.

Henceforth this objective also concerns with the conceptual design of intelligent behaviours in healthcare interoperability, aiming to improve the availability, reliability, time response and quality of the information exchanged among heterogeneous systems.

Deriving from this objective another line of research emerged concerning the moral and ethical concerns when implementing intelligent agents in healthcare environments. Is it ethical to allow intelligent multi-agent systems to learn and interact with the healthcare environment, where small actions can sometimes have complex moral weight?

In a healthcare unit, intelligent agents can be used as a mean towards the integration of different services and the software being used. Within this system, different intelligent agents, autonomously and adaptively, defend individually or by means of cooperation their interests and objectives. They concentrate vital functions of the healthcare unit, improving the quality-of-service and the people quality-of-life. As part of this system there exists different agents which, by different forms, support medical research, having the capacity to interact with its environment and evolve, acquiring new methodologies and information to improve their own qualities and competence, i.e. to solve different problems according to its duties.

For example, a physician, when analysing an exam received from a computerised tomography, is presented instead of thousand of pictures, a smaller number of pictures selected by intelligent agents. In light of the selected images, it was not possible to detect any anomaly. Meanwhile, in the group of selected images missed a small set of pictures which evidentiated the existence of small metastasis which might have changed the diagnose. This case was misdiagnosed by influence of the agents. The physician made a decision which ultimately had moral and legal consequences. This decision revealed itself as a bad help to the diagnose, placing at stake a human life.

Bearing in mind this hypothetic case, it is understandable the relevance of studying moral reasoning and the implications of the introduction of intelligent autonomous agents in such an ethically complex environment as healthcare. From this research question the it is proposed a model in which we aim to ensure the moral reasoning of such agents is trustful.

Summarized the deliverables

- state of the art analysis of interoperability and healthcare informatics

- model for ethical reasoning for agents in healthcare

Prototype platform of the archetype

This objective centers in the development of a modular agent oriented implementation of the core principles that were specified int the previously proposed archetype.

Summarized deliverables:

- technical description of the model
- detail the connectivity between modules and used technology
- basic software structure/code for the easy development of the proposed archetype

Testing and implementation of the archetype

With the prototype developed previously there is the opportunity to implement it in both controlled and production environments. The objective of such task is to determine that the technical development of such model in real healthcare environments can be achieved. For this objective and the subsequent evaluation that is the next objective, the participation with distinct portuguese healthcare institutions is unequivocally a great asset. It allows not only a better understanding of the problem at study and a unique opportunity to devise and implement solutions based on the previously proposed archetype.

Summarized deliverables:

- implementation of the prototype in production healthcare environment
- implementation case studies

Evaluation of the archetype and its implementations

The validation and study of the result of the implemented models is of the essence to evaluate the proposed archetype.

Summarized deliverables:

- indicators regarding the implementation
- conclusions regarding the characteristics of the archetype

1.3 Document Structure

The document structure follows in a non strict manner the proposed objectives in five main chapters. The current Chapter 1 is an introduction to the objective of this thesis, addressing the found research opportunities and studied topics as well as the motivations guiding this research project. It also details the main deliverables to be expected from each objective.

The following chapter (Chapter 2) provides a state of the art review of the main topics addressed in this research. This structure reflects a deeper understanding of the topics mentioned in the introduction so a better understanding of these areas is achieved by the reader. Henceforth, it centers in core topics namely healthcare information systems, interoperability and agent systems. The information detailed in this chapter also intends to introduce the following one, in which the proposed archetype itself is devised.

As mentioned previously Chapter 3 presents the archetype from distinct points of interest. Firstly it perform a requisite analysis so the characteristics the underlying model must have are defined in a clear manner. It is followed by a technical definitions of the used paradigms and software architecture and technologies used and how they intertwine. Afterwards there is a detailed description of the conceptual definition of the core agents and services within the model, which is followed by a structured analysis of a implementation using this archetype for healthcare interoperability.

In order to detail the results of this the devised model and its impact on healthcare interoperability, several published articles that address such implementations and concepts are contained in this chapter.

Chapter 4, as the last chapter is where a general discussion and reflection over the main conclusions are presented. It sums up and assessed the results over the proposed objectives.

Chapter 2
State of the Art

2.1 Health Information Systems

Considerable investments are being taken by major healthcare institutions to develop a proper Health Information System (HIS) that enables interoperability; allows freedom on deploying new IS and disable obsolete IS; consolidates and disseminates medical information; increases the reliability and availability of the services provided; uses unequivocal and standard medical terminology; provides adaptability to support changes of the specific workflows or knowledge representation models; and availability of information and knowledge at the time and to the individual that needs it. (Zaleski, 2009) (Bodart et al, 2000) (Haux, 2006)

The growing development of HIS is related with the pervasive and omnipresent concern regarding the improvement of the services provided in healthcare and diminish the elevated costs associated with these services by optimising the existing resources. (Haux, 2006) An health information system is conceptually the core tool shared among all solutions that exist in the distinct areas of healthcare units. By different words, an HIS can be defined as an information system for the processing and exchange of data, information and knowledge within an healthcare environment. A more practical manner to explain and HIS is as the combined and integrated effort by existing information system to collect, process, report and use information and knowledge within an healthcare environment, to influence decision and management policies, health programs, teaching, research and medical practice within an healthcare unit. (Kirsh, 2008) It is aimed not only to treat the information relative to the patient, but also to ease the extraction of management and clinic indicators, which allow administration and clinic staff to make decisions based in information with quality and improving the planing of programs directed towards the improvement in healthcare practices.

Henceforth, an HIS connects horizontally any services and applications existing in an healthcare unit, requiring interoperation among these distinct services and the integration of what they contain. It must be clear that an HIS should be considered as the combined structure of all existing services and applications, which incorporates all the information available at a given moment within the healthcare unit. (Zaleski, 2009) For this reason, it is the complete set of integrated services that define the essence of an HIS, not only a centralised core application that concentrates into itself some of these functionalities.

Bearing in mind the elevated costs to implement and establish a new HIS, the World Health Organisation (WHO) proposes to every healthcare institution a list of basilar steps for the proper development of such an environment and architecture:

- identify the objectives of the project, the wanted results and the functional requirements for the information system to be developed;
- the institutions information management team must define with technical detail the application necessities, specify the architecture requirements and the main information technologies to be used and determine how these information technologies will be implemented;
- all intervenient with interests in the institution must be included in the project in order to ensure that the system will be adapted towards the needs of all of those that interact with it or depend of it;
- the team must assume the compromise of a project on the long run, with the appropriate time frame defined and the appropriate funding.

Healthcare and its underlying services are constantly in improvement and change, not only in clinical terms, but concerning the used information technologies as well. The documentation based on paper is being gradually abandoned, while computers and new technologies are being included in the functioning of the distinct services of healthcare institutions. For this reason, the capacity of information technologies to allow easy access to all structured information of an patient and present the proper medical knowledge as support for the decision making process, is a predominant concern of an HIS. Presently, a new challenge proposes the change of paradigm into a service orientated towards the customer, contrary to being orientated towards the institution (i.e.: clinicians, nurses, technicians, ...). This a new elemental characteristic, if not vital, to any HIS. (Haux, 2006)

Concerning the implementation of an HIS on such an environment, the Public Health Informatics Institute (PHII) developed a set of orientation lines that define the procedures and postures that a multi-disciplinary team responsible for the implementation of such a project ought adopt: (PHII, 2004)

- approach al interested parties - all parties involved and interested in the HIS must be represented in its development program (e.g. administrators, informations systems experts, physicians, physicians, nurses, technicians, ...);
- create models before acting - understand the process of defining the requisites of an healthcare institution before its physical implementation or development of any component of the system. This process is the most important in the establishment and bought of an HIS;

- plan for interoperability- plan the communication and integration of different systems within the healthcare institution, This step is of the essence of the modularity and scalability of the architecture in which the HIS services append on;
- manage responsibilities - the analysis of the capacity of the system to respond and match the necessities of all users, the necessity of funding and consider the timeframe available for the completion of the project, is essential for the good unroll of the project.

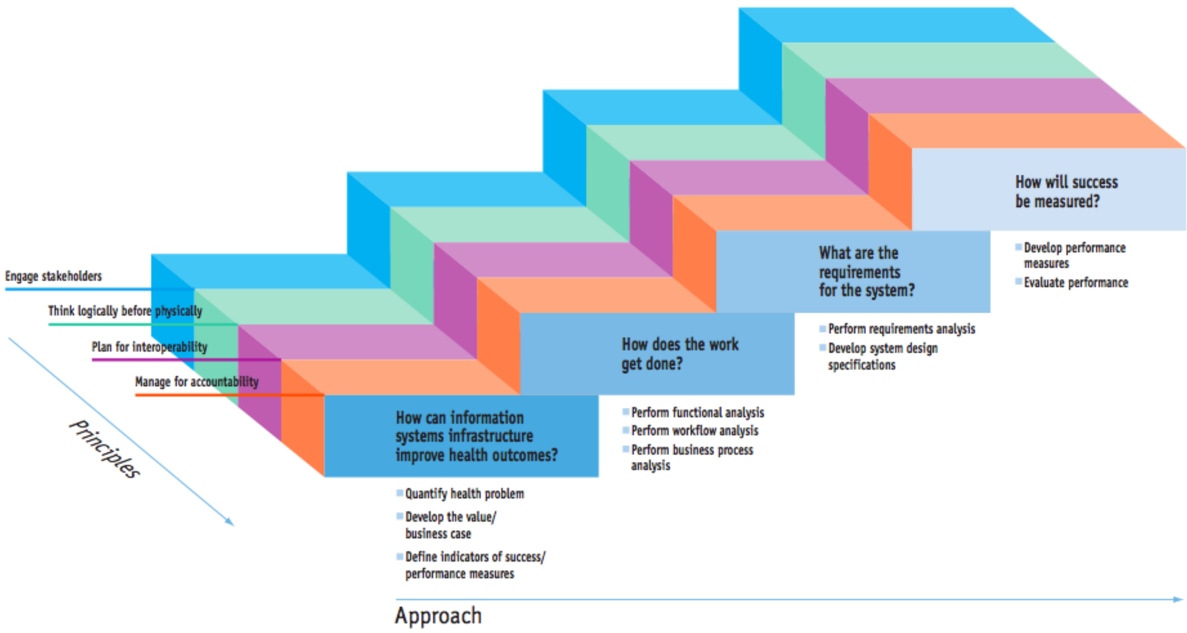


Fig. 2.1: Analysis of the guidelines proposed for the implementation of an HIS

For the success of the implementation of an HIS on the, it must have the capacity to adapt to the constant evolutions of medical practices, as well as predicted alterations or proposals of workflows in the functioning of the different services that are part of the healthcare institution. A very important factor to bear in mind in this situation is the intervention of all parties interested in the system, i.e. beyond having attention to the necessities for the administrative and technical component associated to this information systems, it is vital to consider the need and opinions of clinical staff. (Berg, 2004) The fact that is very common if no inherent, the existence of extremely distinct services withins the healthcare arena and their extremely specific softwares, poses a question of interoperability and the development of methodologies for the integration in order to overcome the heterogeneity within every HIS.

2.2 Integration and Interoperability

From this need for independent information systems to communicate and cooperate in order to enhance their overall performance and usefulness, the notions of integration and interoperation were introduced at different conceptual levels with distinct objectives, but with intersecting principles. Both this principles are important for cooperation and the flow of information within and between organizational units, however they are based on disconnected principles. In simpler terms, integration aims to gather and acquire information of distinct systems into another that requires such information, while interoperation centres on the continuous communication and exchange of information between cooperative systems. A more distinct idea can be achieved thought the understanding of their individual definition and boundaries.

The Institute of Electrical and Electronics Engineers (IEEE) defines interoperability as the "ability of a system or a product to work with other systems or products without special effort on the part of the customer". It also states that interoperability is made possible by the implementation of standards. In a similar but more complete manner, according to the International Organisation for Standardisation (ISO), interoperability is the ability of independent systems to exchange meaningful information and initiate actions from each other, in order to operate together to mutual benefit. In particular, it envisages the ability for loosely-coupled independent systems to be able to collaborate and communicate.

2.2.1 Principles of interoperability

The intrinsic characteristics and methodologies for problem solving under an interoperable setting are complex to be determined in one single and absolute model, as numerous models can be found that are both valid and sound. These models attempt to classify interoperation approaches and activities using one or sets of attributes, which define the exchanged data abstraction level, technological implementation, interoperation viewpoint and underlying purpose.(Initiative, 2010)

The Dublin Core Metadata Initiative (DCMI), an organisation that aims to provide simple standards to facilitate the finding, sharing and management of information, created an interoperable framework at the system level, which is mainly based on the abstraction level of the exchangeable data or records, which is to be understood in terms of 4 (four) tiers, namely:

- Level 1 - Shared Terms Definitions - data components with shared natural language definitions;
- Level 2 - Formal Semantics of Interoperability - data is based on formal-semantics;
- Level 3 - Description Set of Syntactic Interoperability - data is structured according to shared formal vocabularies in exchangeable records; and

Table 2.1: Implication of LMCIM (adapted from Tolk and Muguira (2003))

Level	Designation	Information abstraction level	Information defined	Contents clearly defined	Domain	Focus
L6	Conceptual	Common conceptual model	Assumptions or constraints	Documented conceptual models	Modelling abstraction	Compositionally
	Dynamic	Common execution model	Effect of data	Effects of information exchange		

2.2.1.1 Health Level Seven

L5						
L4	Pragmatic	Common workflow model	Use of data	Context of information exchange	Simulation implementation	Interoperability
L3	Semantic	Common reference model	Meaning of data	Content of information exchange		
L2	Syntactic	Common data structure	Structured data	Format of information exchanged		
L1	Technical	Common communication protocol	Binary data	Symbols of information exchanged	Network connectivity	Integrability
L0	No	No connection	NA	NA		

- Level 4 - Description Set Profile Interoperability - data content is structured according to shared formal vocabularies, being bounded by a set of invariants on the exchangeable data or records.

These levels are oriented to the Dublin Core environment and its reference model, but can be extrapolated to other metadata models.

Another model proposed by Tolk an Muguira aims to divide the conceptual layers of the interoperable factors according to the exchanged data abstraction levels, technological implementation and underlying purpose/focus. These 7 (seven) layers, as it is depicted in Table 2.1, ranging from level L0 to L6, define a scenario from no interoperability to a scenario with conceptual interoperability. In this model not only abstract concepts but also methodologies for problem solving are grouped according to their impact and potential to the interoperation process. (Tolk and Muguira, 2003)

A distinctive description adapted by Mykkänen and Tuomainen (Table 2.2) for evaluating and classifying interoperation standards and models, is also of interest to the classification of the interoperable levels in itself. Although for more oriented approaches towards the identification of which interoperable aspects are not covered by a standard (i.e. from level 1 to 7), due to this underlying capability, it describes a model that put under a different perspective the interoperability process. (Mykkänen and Tuomainen, 2008)

Table 2.2: Interoperability levels (adapted from Mykkänen and Tuomainen (2008)

Level	Designation	Meaning
7	Application life cycle interfaces	The life cycle of the application, including integration and development methods
6	Functional reference model	The domain-specific information or functional model or assumptions about the used methods
5	Semantics	The meaning of the defined interface elements
4	Functional interfaces	The defined functionality and information
3	Application infrastructure	The integration points in the distribution architecture of the participating applications
2	Technical infrastructure	The infrastructure for supporting the interface and communication technologies
1	Technical interfaces	The technologies used in the interfaces and implementations

Far from comparing these models, several tendencies in the abstraction level of the exchanged information and the usages that are made of such information may be detected. One distinct borderline is the notion of syntactic and semantic level of information. In all models a clear difference there exists in the definition of shared syntactic rules, and shared semantic meaning and relations of the exchanged information. By setting shared syntactic rules, the content of all information within the exchanged data is normalised in such a way that it is possible to determine the nature of its content. However, only with shared semantic meaning and relations can the content be understood by the existing systems, enabling them to be connected with their equals, and at the same time to validate it according to the semantic relationships already defined.

The validation of information conforming to shared constraints is presented in the final levels of both the LMCIM and the DCMI, being given as the main characteristic defining the final level for interoperability. Indeed, the validation of exchanged data and rectification of incoherences is an important feature. Moreover, the use of shared constraints and formal vocabulary requires prior shared semantic models, something clearly stated in both models.

These concepts of interoperability are essential to understand and explore the particularities of interoperability in healthcare, where high levels of interoperation and exchange of information are required. These requirements are defined either by this environment or tools, such as the Electronic Health Record and the Decision Support Systems that are being integrated into it.

2.2.2 Interoperability in healthcare

Although interoperability had been studied and its implications to care delivery had been considered, the level of interoperability among systems in most healthcare institutions remains frustratingly low. (Carr and Moore, 2003)

In 2005 ISO released the ISO Health Informatics Profiling Framework (ISO TR 17119:2005) (ISO/TC215, 2005), a vehicle tailored to describe standard artefacts, i.e. one of many kinds of tangible byproducts produced during the development of software in a healthcare environment. Similarly to other basic structures underlying a system, it aims to detail, classify and create relations among items within the domain area. This framework place such instruments under six perspectives, namely what, how, where, who, when and why, and three levels of specificity, that is to say conceptual, logical and physical design. This framework is oriented to organise and direct the development of a level of quality or attainment in the area, however it seems not to consider the existent norms such as Health Level Seven (HL7) and Digital Imaging and Communications in Medicine, which are decisive in most of existing interoperation processes in healthcare.

An intensive effort to develop standards adapted and optimised towards healthcare delivery had been developed before the ISO framework, and apparently resulted in more than two but not many consensually models in comparative evaluations. These standards have been able to give a definite structure or shape to low level interoperability in healthcare, in a firmly established and modular manner. Among these patterns HL7 is considered the most adaptable one in healthcare interoperability.

2.2.3 Health Level Seven

HL7 started as a mainly syntactic healthcare oriented communication protocol at the application layer, the seventh layer of the OSI communication model. This protocol defined the message structure to be exchanged by loosely connected healthcare applications by classifying the different types of messages involved in this environment with the aggregation of standardised segments.

The structuring and design of this standard, defining which artifacts of data should be transferred by a certain message, enabled and potentiated the application of HL7 in client-server architecture. (Ohe and Kaihara, 1996) The most common implementation of this architecture using HL7 is based on distinct socket communication clients and servers, in which the client sends an HL7 structured message to the server, that upon processing sends an acknowledgement HL7 standardised message. The HL7 standard is not bound to this architecture, but it is the most widely used in healthcare interoperability.

In regards to version two of HL7, its structure is very simple and can be defined technically as a conjunction of ASCII lines separated by a carriage line return, and in which the first line contains the control section, the MSH segment that defines the structure of the message itself.

In the control section of the HL7 standard several rules that are applied to all messages are defined

- Message Segment
- Message Type
- Tigger Events

The communication design flow of HL7 is centred on the events approach, as a model of the events that occur in the non-digital world. This mimicking of hospital processes and concepts is an important characteristic of this standard. Its major requisite is indeed the data to flow among heterogeneous systems that comprise the HIS, and the event paradigm an answer to the usually asynchronous and complex chain of messages exchanged physically and digitally within healthcare institutions.

According to this paradigm, most events in the healthcare environment are modelled and act as triggers for the initiation of data/information dissemination. This event can emerge either by interface or back-end system, however its asynchronism can be demanding of interface behaviour. Regarding the actual steam of information, there are distinct levels of event handling. In the one hand, an event can be received at one specific system and be handled by this system alone, being the flow of information to other ones aimed mostly at maintenance of consistency. On the other hand, an event can be initiate at one system but need to be handled by another, in which case the information transaction is named an unsolicited update.

The understanding of the event handling is of the essence to model HL7 interoperability work-flows in order to properly understand the dependencies among systems and data.

The scope of the standards aim is to solely specify messages between systems and the events which trigger them. No considerations regarding underlying systems architecture and implementation are concerned by HL7.

A trigger event may come from one of the following sources:

- User Request Based - For example, the trigger event that prompts a system to send all accumulated data to a tracking system every 12 hours is considered Environmental. Similarly a user pressing a button in a user-interface would be considered environmental.
- State Transition - resulting from a state transition as depicted in the State Transition Model for a particular message interaction. The trigger for cancelling a document, for example, may be considered a State Transition Based trigger event.
- Interaction Based - based on the receipt of another interaction. For example, the response to a query (which is an interaction) is an Interaction Based trigger event.

The sources of triggering events here detailed model the usual behaviour of agents, being it human through interface interactions or autonomous/semi-autonomous information systems. Although the message types are not strictly associated to an event source, the source of one event does in an intrinsic manner define the event itself and henceforth the message that it should generate. Non withstanding this intuitive notion, the uses of the same message type to different sources is common, such as the usage of the message type ORM-O01 to make a general order message that is used to transmit information about an order. Regading this message, there is only one type of ORM messages – the ORM-O01 message. Trigger events for the ORM-O01 message involve changes to an order such as new orders, cancellations, information updates, discontinuation. They it is among the most widely used message types in the HL7 standard.

This initial versions of HL7 is uniquely syntactic, and according to the general models of interoperation are one of the lowest levels of this process. The current version 3 is opening the HL7 scope towards semantic interoperability, including the appropriate use of exchanged information in the sense of the communicating applications behaviour. This model presented in version 3 contains relations and metadata in a abstract level that may enable far higher levels of integration, namely by semantic interoperability and validation of exchanged information, using the relational mapping of each artefact. The Message Development Framework (MDF) is currently moving towards the HL7 Development Framework (HDF), therefore shifting the HL7 paradigm from message to architecture. Newer HL7 developments such as the EHR-S Functional Model and the SOA Project Group activities have been pushing this move (Lopez and Blobel, 2009).

The metadata and archetypes defined in HL7 allow it to organise both production and clinical data in clearly defined and connected segments and fields, which can be validated among artefacts. However, the implementation of version 3 is still rather limited as few service providers and institutions migrated already to this version.

2.2.4 Interoperability towards an unified electronic medical record

The Electronic Medical Record (EMR) is a core application which covers horizontally the health care unit and makes possible a transverse analysis of medical records along the services, units or treated pathologies, bringing to the healthcare arena new methodologies for problem solving, computational models, technologies and tools. This application must be centered on the patient under the different perspectives of the medical staff.

The healthcare arena configures an environment where numerous specific solutions store in independent data structures the information of the patient, production and other significant data. Due to the complexity of each of the inner arenas of healthcare the possibility of an global information systems emerges as overall complex and incomplete, as the effort to cover all services provided is overwhelming as incomplete will be this attempt compared to a specialised application. However the need to gather the significant information to be shared to other services and to communicate all relevant data related to the patient and the executed procedures is of high value not only to the institutions, but also to the patient.

In order to aggregate and consolidate all significant information information solid and efficient processes of interoperation or integration must be developed. These processes must take into consideration scalability, flexibility portability and security when applied to EHR. The underlying EHR architecture must be based on the component paradigm and model driven, separating platform-independent and platform-specific models.(Blobel, 2006)

The complexity and sensibility of the exchanged information requires more than technological efficiency and pragmatic exchange of information. The dissemination of incoherent information and its introduction into the EHR may cause more than inconsistent records, they may be the base for a misdiagnose of bad choice of medical practices. For this moral and ethical complexity to be avoided a thorough validation of the exchanged and integrated information must be performed. The development of top level interoperability frameworks is henceforth of the essence for the healthcare environment. The multitude and intricacy of services that must be performed by the EHR and Group Decision Support Systems (GDSS), require such a framework or otherwise would be inefficiently intertwined with other essential solutions. (Miranda et al, 2009a) (Duarte et al, 2009)

2.3 Agent Based Systems

In this section it is described in greater detail the concept of multi-agent systems and the usage of agents as a core approach towards artificial intelligence.

2.3.1 Notion of agency

The key principle of software agents is that it is an entity which receives information from its environment and has the ability to actuate in this environment in a manner than can change it in some way (Jennings et al, 1998). This environment can be either physical or virtual, implying that the interaction with it can be also abstract as is the case of supporting the decision support. Henceforth, an agent as percepts its reality in its particular manner depending

When the paradigm of agents emerged there were complication in defining it, resulting in noise and confusion about it. In Wooldridge and Jennings (1995a) there is a work on the authors to find a more concise definition. In further detail, Wooldridge and Jennings segments the notion of an agent in a *weak notion of agency* and a strong notion of agency. The weak notion of agency is a most general manner of describing a hardware of more usually a software-based system that enjoys the following characteristics:

- **autonomy**
- **social ability**
- **reactivity**
- **pro-activeness**

Autonomy

Agents are able to function without the direct intervention of human or other genre intervener, and have some kind of control over their actions and internal state. They are able to encapsulate theirs state, which is not accessible to other agents, and from which make decisions about what to do based solely on this state and their possible ways of interaction, without any direct intervention from other entity.

Social ability

agents interact with other agents (and possibly humans) via some kind of agent-communication language, and typically have the ability to engage in social activities (such as cooperative problem solving or negotiation) in order to achieve their goals.

Reactivity

One of the core characteristics of agents mentioned before is that they are situated in an environment, which may be the physical world, a user via a graphical user interface, a collection of other agents, the global network, or perhaps many of these combined. They are able to perceive this environment (through the use of potentially imperfect sensors), and are able to respond in a timely fashion to changes that occur in it.

Pro-activeness

agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative;

This approach is very simple and aims to define the basic core capabilities of agency in order to be determined as one. However most researchers attribute stronger and more specific attributes which may differ from the underlying aims and premisses of the agent theory involved. These concepts were labelled as *stronger notion of agency* (Wooldridge and Jennings, 1995a).

2.3.2 Agent Architectures

With the establishment of the definition of agent and the theoretical views of what an agency should embody, distinct methodologies emerged as research trends related with the agent community. This means that agent architecture above all are software design methodologies which reflect the collective knowledge about what methodological approaches are best adequate in order to develop an instance of intelligence. Due to the distinctions inherent in theseThere are numerous manners of segmenting agent architectures,(Bryson, 2000).

This section segments and classifies the most common agent architectures as was inspired by the paper of (Müller, 1997; Cranefield and Purvis, 1996) into Reactive, Deliberative, Interactive and Hybrid. This knowledge is important to understand the present implementation of agent based systems and what is gained or lost with each of the architectural choices made during that process.

2.3.2.1 Reactive Agent Architectures

Reactive intelligence is the idea that controls a reactive agent, a model of procedure which can respond very quickly to any changes to variables which comprise the environment it perceives.This architecture was highly influenced by the behaviourist psychology, therefore their denomination as behaviour-based, situated or reactive. Their reasoning is usually based on a rather limited amount of information and perception of their environment, associated with simple situation-action rules, which are processed at run-time (Müller, 1997; Cranefield and Purvis, 1996). Although this characteristics and genre of intelligence have been associated with statelessness, this perception is exaggerated as every intelligent or autonomous systems requires a notion of state for learning, adaptability, perception and complex control (Bryson, 2000). Regardless of having state it still comprises minimal representations and minor deliberation.

With Brooks *subsumption architecture* the need for a sub-symbolic representation of the world was denied, defining that agents would make decisions directly and uniquely from their sensory output.

The representation of reactive behaviours is aimed towards obtaining a overall robust behaviour rather than a correct representation of the world and associated best behaviour. Their robustness derives from the ability to predict and control otherwise complex lines of action.

2.3.2.2 Deliberative Agent Architectures

In Newell and Simon (1976) it is implied that agents maintain an internal representation of their environment in conjunction with an model of explicit mental state that can be modified internally by some kind of symbolic reasoning. This internal reasoning process is a characteristic that is core for deliberative agents. In other words these agents possess and intrinsic image of the external environment and are therefore capable to control or plan its own actions according to their reasoning an not direct external stimuli.

One of the most prominent architectures that derives from the deliberative architecture models agents in their beliefs, desires and intention, being therefore known as the DBI architecture. These models defined the concept of agency using the anthropomorphic notions of metal states and actions (Mora et al, 1999)(Georgeff et al, 1999). This architecture was firstly formalized by Rao and Georgeff (1995) by defining the concepts of belief, desire and intention in association with the underlying logic, namely the notion of desires as model operators.

Henceforth, as the name suggests the internal state of an agent under this model is comprised of three key data structure, which in a loosely manner aim to to correspond to beliefs, desires and intentions. These states as mentioned previously are internal and only manage by the agent entity itself. An agent's beliefs are a representation of the information it perceives of the environment that surrounds it. These beliefs are usually represented symbolically in the Procedural Reasoning System and are very alike PROLOG facts (Wooldridge, 1997) (Padgham and Lambrix, 2005)(Georgeff et al, 1999). The desires of an agent can be understood as the tasks made available and that are allocated to it. Finally and agent's intention represents desires that it has internally committed to achieve. According to Wooldridge (1997) understanding of Rao and Georgeff definition of the model, an agent will not, in general be able to achieve successfully all of its desires. Henceforth it must reason which subset of desires it will have achieved and allocate resources necessary it order to do it. This subset of desires are then intentions and will thenceforth be used as feedback into posterior decision making processes.

The operators made available to model the BDI architecture are of great significant in order not to create a gap between the the agents systems that are developed and their corresponding natural model. As it is mentioned by Mora et al (1999) there are two major techniques that try to address this issue. One attempts to extend BDI logic with the necessary operational models so that the agent theories become computational. The other is based on suitable logical formalism in that are both powerful to represent the mental states of the agents and that have operational procedures that allow the use of logic as a knowledge representation formalism (Padgham and Lambrix, 2005)(Georgeff et al, 1999).

Currently there are several agent frameworks that formalize BDI architecture either by the providing extensive operational models or using logic programming, being some of the most popular:

- Jadex
- Jason
- Goal

2.3.2.3 Interactive Agent Architectures

An important feature of agents when in a distributed environment is the ability of argumentation and cooperation among intelligent agents. The frameworks that implement and interactive architecture incorporated copperative habilities onto the core of its behaviour.

InFischer (1993) and Fischer and Fischer (1993) it is detailed one of the first implementations of such architecture, MAGSY. It was basically a language for the design of multi-agent systems in a fairly simple matter in which agents ca easily provide services to one another. The core reasoning was based on a set of facts that represented its core knowledge, a set of rules that represent the agent's strategies and behaviours and a set of services that design the agents interface for request and processing of services.

MAGSY provides a a variety of useful services and protocols to establish multi-agent communication links. Although these services and paradigm are a step towards cooperative agents it inherited both the positive a negative sides of rule-based programming. Although is very simple suitable to represent reactive agents and clean manners to deal with concurrency, there is the flat knowledge representation and the hardwired manner to represent sequential programs. Henceforth the connections between gents a re hard-wired and represent the behaviors of each of the different agents, not providing full fledge cooperations

2.3.2.4 Multi-agent systems

The agents systems that were composed of more than one agents were part of the research of the field of Distributed Artificial Intelligence, which was divided in two main fields, Distributed Problem Solving and Multi-Agent Systems. More recently these term Muti-Agent System has ben associated to all types of software systems that are composed of multiple semi-autonomous components (Wooldridge, 2009) (Jennings et al, 1998).

The focus concern of a MAS is the collective behaviour of a set of existing agent entities that may or not be working towards solving a given problem. In (Jennings et al, 1998) MAS is as a loosely coupled network of problem solvers that work together towards solving problems that are beyond the capabilities of a single instance of one of them isolated.

In an abstract manner these entities that can be defined as agents are autonomous and heterogeneous in nature. This characteristic is very important to consider given the objective of this thesis. The distribution of decision-making among a large number of autonomous agents has both advantages and disadvantages. Most of the disadvantages arise from emergent or complex behaviors, which are difficult to predict specially considering they can be executed in parallel and involve agent argumentation and underlying reasoning. These characteristics can be complex to predict and control, while they may contain as well unpleasant surprises. On the other hand many advantages result from this paradigm (Talukdar, 2004):

- Parallel processing: The difficulty of many problems grows exponentially with their size. One way to handle such problems is to decompose them into smaller problems that can be solved in parallel.
- Scalability: The multi-agent paradigm allows to add and remove agents according the the environment needs and requirements.
- High speed: The reflexes of a system can be quickened by arranging for the agent-on-the-spot to make decisions, without the delays involved in dealing with a long chain-of-command.
- Increased robustness: When an autonomous agent has issues and subsides to exist, another with similar internal functions can compensate for some, if not all, of its loss.
- Cooperative: their core characteristic is argumentation and communication, while working together for a goal.

Although there has bee several studies defending the usage of multi-agent systems in problems of systems that need to exchange information with each other is a robust manner, there are still some issues regarding external attacks to individual agents (Zhang et al, 2012). The early systems that were developed in closed networks composed a trustworthy environments, in which the entities were reliable and collaborative. If one considers an environment as the world-wide-web we find a distributed and heterogeneous environment, in which agents have to live and interact with strangers, which may have undermining intentions(Cavalcante et al, 2012). This is a major factor of concern when developing agents systems in which you don't control every intervenient.

When agents interact it is important to understand their dependencies and how the influence one another.The basic idea is that a dependence relation exists between two agents if one of the agents requires the other m order to achieve one of its goals. There are a number of possible dependency relations (Wooldridge, 2009):

- Independence - There is no dependency between the agents.
- Unilateral - One agent depends on the other, but not vice versa.
- Mutual - Both agents depend on each other with respect to the same goal.
- Reciprocal dependence - Both depend on each other for some goal, just not necessarily the same.

These types of relations are very important to be understood when modelling a multi-agent system, specially in the case of interoperability, as is the case of the systems that are associated to the research under this thesis.

Multi-agent system present characteristics that are very adaptable to the requirements of interoperability and integration. Henceforth the selection of this paradigm as a tool to solve many of the research opportunities and associated challenges found during this research.

2.4 Agent morality in healhcare

Ethics is focused on moral goods rather than natural goods. However, both moral and natural goods are equally relevant and have to be taken under consideration. Morals are created by and define society, philosophy, religion or individual conscience, usually associated with the fundamental questions concerning the complexities of the human soul (Deigh, 1992). Several forms of ethics have been approached, namely the ones:

- Applied ethics, i.e. ethics seeks to address questions such as how a moral outcome can be achieved in a specific situation;
- Normative ethics, i.e. how moral values should be determined;
- Descriptive ethics, i.e. what morals people actually abide by;
- meta-ethics, i.e. what the fundamental nature of ethics or morality is, including whether it has any objective justification; and
- Moral psychology, i.e. how moral capacity or moral agency develops and what its nature is.

The role of computers is rapidly evolving from that of passive cipher to that of active participants in the trading process, which lead us to an imperious need of analysing the questions of morality. In Philosophy, morality has different meanings, namely (Deigh, 1996):

- A code of conduct which is held to be authoritative in matters of right and wrong;
- An ideal code of conduct, one which would be espoused in preference to alternatives by all rational people, under specified conditions; and
- A synonymous of ethics, the systematic philosophical study of the moral domain.

On the other hand, interoperabiliy in healthcare units is defined as the ability to move clinical data from place to place. Bringing interoperabity to these facilities it is possible to reduce costs and give to clinical and medical staff more powerful tools for patient assistance, in particular in the decision support and problem solving procedures. In Medicine, physicians and nurses have daily to deal with incomplete information, which in association with moral judgements and emotivism, turn decisions sometimes wrong, slow, expensive or unacceptable. This leads us to the need of defining formalisms to identify and evaluate morality and ethics in Medicine.

Medical ethics is primarily a field of applied ethics, the study of moral values and judgments as they apply to Medicine, in particular the examination of particular issues that are matters of moral judgments and morally correct behavior in various fields. Medical ethics encompasses its practical application in clinical settings as well as work on the fields of History, Philosophy, Theology, and Sociology. Medical ethics tends to be understood narrowly as an applied professional ethics, whereas bioethics appears to have worked more expansive concerns, touching upon the philosophy of science and the critique of biotechnology. The two fields often overlap and the distinction is more a matter of style than professional consensus. Medical ethics shares many principles with other branches of healthcare ethics, such as nursing ethics. Some attributes that may apply to Medical Ethics are depicted below (Deigh, 1996):

- Autonomy,i.e. the patient has the right to refuse or choose their treatment;
- Beneficence, i.e. a practitioner should act in the best interest of the patient;
- Non-maleficence, i.e. "first, do no harm";
- Justice, i.e. concerns the distribution of scarce health resources, and the decision of who gets what treatment;
- Dignity, i.e. the patient (and the person treating the patient) have the right to dignity; and
- Truthfulness and honesty, i.e. the concept of informed consent has increased in importance in the last few years.

Those parameters must be quantified and its importance can not be sub-estimated in the decision making process. All the ethical questions around virtual entities or agents, have to be taken under a practical perspective and are related with the embedded environment. This study has been performed before in terms of electronic commerce, considering the case of the legal and ethical context of contract made by means of intelligent agents (Andrade et al, 2004) (Andrade et al, 2005). Nonetheless, there exists the need to undergo a particular approach when considering morally dubious areas, where every little may have great moral consequences. This is the case of Medicine, where interoperability and decision support are presently in continuous analysis and development. Following this thread of thought and taking in consideration the state of the art of the Agent Oriented Paradigm, it will be analyzed in this study the moral context of agents, discussing the possibility of an agent at a given state of development, have the moral capacity and legal responsibility for actions.

2.4.1 Intelligent agents and medical ethics

In a healthcare unit, intelligent agents can be used as a mean towards the integration of different services and the software being used. Within this system, different intelligent agents, autonomously and adaptively, defend individually or by means of cooperation their interests and objectives. They concentrate vital functions of the healthcare unit, improving the quality-of-service and the people quality-of-life. As part of this system there exists different agents which, by different forms, support the medical research, having the capacity to interact with its environment and evolve, acquiring new methodologies and information to improve their own qualities and competence, i.e. to solve different problems according to its duties.

For example, a physician, when analyzing an exam received from a computerized tomography, is presented instead of thousand of pictures, a smaller number of pictures selected by intelligent agents. In light of the selected images, it was not possible to detect any anomaly. Meanwhile, in the group of selected images missed a small set of pictures which evidentiated the existence of small metastasis which might have changed the diagnose. This case was misdiagnosed by influence of the agents. The physician take a decision which ultimately had moral and legal consequences. This decision revealed itself as a bad help to the diagnose, placing at stake a human life. Another important topic in medical ethics is the concept of futility. What should be done if there is no chance that a patient will survive but the family members insist on advanced care? And what should be made if a patient is in a Intensive Care Unit, using a bed that is necessary to save another patient with more chance to survive? Rational decisions can be taken to solve this particular problem, following legal or practical rules, either by physicians or by intelligent agents. But who will be responsible for taking such moral decisions? Facing such situations, several questions and doubts arise, namely: What is or defines a Moral Agent? Is an intelligent agent a Moral Agent? Will these agents have at any point in time either the capacity and ability to take moral decisions or being capable to handle with decisions which carry a great ethical dilemma? Which are the legal and moral responsibilities in an agent based system?

The present period, or step in a process or development in Artificial Intelligence (AI) is still far from the usual scenarios imagined by science fiction. However, it is becoming an embedded characteristic in application of different areas, from Commerce to Medicine. Indeed, AI techniques which imbue software systems with a considerable degree of intelligence, autonomy and proactivity, and the ability to adapt to the environment being populated are growing, being essential to attain a superior level of utility and interactivity. In fact, it urges the necessity to evaluate and regulate the scope of the capacities of this software systems, either when they are called to execute different tasks or to take decision which may have any arguable moral value. The field of ethics associated to non-organic entities, Machine Ethics, thereby lacks of a more practical oriented and cautious reflexion, that will analyze the state of the art of AI in all its vast extension. It will be then possible to defined moral competences and restrictions to its use in any environment, where morality and reputation are to be questionable.

2.4.2 Intelligent agents in medicine

The requirements of software applications for the healthcare arena, although being rather similar to those of other areas, develops in a completely different dimension due to the value inherent to the moral good, i.e. the health condition of a human being. All agents, either human beings or software agents need to be aware of the immeasurable value of an human life and the ethical complexity existing when dealing with this specific good. As information systems continue to disseminate and strengthen in the healthcare sector, the significance of their use increases and so does their moral responsibilities, i.e. a great part of the scope of intervention of agents in this area carries a moral context and ethical responsibility, which it is made aware, even in software artifacts that inevitably will be designed to automatize and manage the larger loading of information generated by medical practices and underlying activities. In fact, this amount of information is so huge that it becomes impracticable to store and extract any sort of knowledge, without the use of computational methods and AI based techniques. From the different computational paradigms in use in AI, Agent Oriented Programing has pursued a sheer growth considering the level and number of the available systems, being capable of integrating other technologies and techniques for problem solving such as Neural Networks or Case Based Reasoning. An individual agent or a network of agents based on different communities of agent possess a class of characteristics that allows them to be independent from the will, desires and objectives of other virtual agents or human beings, granting a certain degree, although limited, of individuality. On the other hand, an agent method cannot be invoked by other than the entity itself, being determined by its will and degree of responsibility. Only the agent is in charge of its own behavior. Regardless, the use of learning techniques from AI, enable the agent to contextualize and evolve dynamically, making the underlying behaviors dependent of the environment, as well as from other circumstances, which may go out of the scope of its initial parameterization. This possibility rises issues concerning the ethical and legal responsibility of the agent owner, in line with the characterization of intelligent agents as autonomous, self-learning and dynamic entities. The distributed and heterogeneous nature of this environment, makes the best use of this technology (Nwana, 1996), which is being applied to different services and situations, going from heterogeneous system integration to decision support systems (Machado et al, 2008). A great effort of academic and corporate synergies allowed the use of intelligent agents in several medical centers which aggregate several hospitals and health units, which use an Agency for Integration, Archive and Diffusion of Medical Information (AIDA), an agent based software artifact, that intends to integrate and agregate information from different systems and locations (Miranda et al, 2009a). On the other hand, the use of intelligent agents for integration of systems may not seem to hold a great deal of ethical significance. However,although these tools improve the security and functionality of the medical information as a whole, the consequences of the loss or adulteration of clinical information or the permissiveness towards this sort of actions, carries a unmeasurable ethical and moral value. A lot more can be said about the decision support systems whose action, although being in support of a decision, contributes indirectly to the diagnose and the treatment of patients. Taken these situations into consideration, it becomes essential an objective discussion about the capacities and characteristics of these systems, in order to define the moral competences of an intelligent agent. This characterization is vital as weell as the need of practical guidelines and rules or ethical conduct for the development and behavior of this sort of systems, so that the quality of the services provided may improve.

2.4.3 Moral capacity

A moral agency is defined by the moral requisites that drive its behavior. In this way, the underlaying concept of a Moral Agent (MA), relies on the existence of moral premisses that rule its behavior, differentiating good from evil. It is important not to misunderstand the concept of MA with Moral Patient (MP). While the first has moral obligations, a MP is an entity in which moral rights speak for themselves. Moral agents are in most of the cases moral patients, however this relation is non-reciprocal, as the discussion on delimitating the grounds of MA considers that only a part of MP are in fact capable of being MA. An adult is a MA although a recently born child is solely a MP, being capable, however, of becoming one during his/her life time (Himma, 2008). This statement that an entity may become during its life time a MA, is indeed very important, once it allows, in an analogous way, to state that an agent, at a given moment, acquire such a property. It is necessary to understand what is a intelligent software agent and which are the characteristics that will allow it to become a MA.

According to Wooldridge, an agent embodies a computational system capable of revealing an autonomous and flexible action, developed in a determined universe of discourse. The flexibility of the agent is related with its capacity of reaction, initiative, learning and socialization (Wooldridge, 1999). Although the definition may not be considered an universal one, for an organic or software based entity, there exits two levels of intrinsic characteristics, which define in a weak or strong form, whether or not that entity is an intelligent agent. On the one hand, the weak notion of agent represents the minima characteristics of an agent, centering in its capacities of autonomy, reactivity, pro-activity and sociability. On the other hand, in the strong notion of agent, are defined imminently cognitive characteristics, that can result in the development of a self-consciousness by part of the agent and in the enablement of other valuable properties such as perception, sentimentality and emotions (Wooldridge and Jennings, 1995b). The establishment of this characteristics is an important factor in the contextualization of the designation of intelligent agents in a way to normalize what is in fact and object, and any other form of software based intelligent entities.

The comprehension of these characteristics has to be a analyzed relatively to a Level of Abstraction (LoA) that uniforms them and limit the possibility of relativism on their analysis. Turing first used the notion of LoA to, according to a level established by him, to define intelligence. This concept was used by Floridi and Sander to analyze, according to different LoA the characteristics of intelligent agents before their capacity to undertake moral decisions. Although LoA is a concept derived from Computer Science, more concretely from the disciplines of Formal Methods, that uses discrete mathematics to specify and analyze the behavior of information systems. A LoA consists in a collection of observable, being each one defined by a group of values and results. In other words, before the same entity there exists different LoA that characterize it in a distinct way without adding any type of relativity in the resulting observation. Given a set of values of X well defined, an observable of the type X is the variable which response values are contained in X. A LoA consists then in a collection of observable that are considered on the observation. In a less abstract level, in the case of a car, there can be defined different LoA such as of the buyer, mechanic, insurance officer, all of which present different points and characteristics that, even being distinct, do not present relativity(Floridi and Sanders, 2004).

Depending on the LoA, just as an entity can be considered an object or an intelligent agent, defining the proper LoA the properties that defined a MA can be be of use, being for this reason the notion of LoA used by Floridy and Sanders to define the criteria which must be included in the LoA of a moral agent. The three criteria considered in this LoA are interactivity (a), autonomy (b) and adaptability (c), being the synchronous existence of these characteristics what enables an intelligent agent to become a MA (Floridi and Sanders, 2004). In order to better analyze these characteristics, one must specify and adequate them with the definition of an agent and, as well, with the state of the art of the development of intelligent agents, namely:

(a)The base for interaction underlaying this study is related with the capacity of the agent to identify and comunicate with other agents, nevertheless their nature, i.e. wether they might be MA or AM, software based or human beings. It can be related with the reactivity described by Wooldridge, before different stimulus provided by the environment where the agent is based. Comparing with the string definition of agent, one can perceive this as the ability to socialize and relate with another.Taking into consideration this property there are norms developed by the Foundation for Intelligent Physical Agents (FIPA) in order to normalize the communication among agents in different systems and based on different technologies (Bellifemine et al, 2007) (Wooldridge and Jennings, 1995b).

(b)The autonomy of an agent is a function of its grasp on the universe of discourse and must be in line with its own objectives. A moral agent has the capability to change its state without any external intervention that will force it into a particular line of action. Contrary to an object in Object Oriented Programming, a moral agent is not invoked and in a certain way "forced" to execute a determined action; the agent only performs actions according to its own directives. This characteristic is already considered so essential that exists middleware based on Object Oriented Programming, like the Java Agent DEvelopment Framework (JADE), that protect their agents from remote evocation Bellifemine et al (2007).

(c) The adaptability of an agent is linked to its capacity to learn and adapt its own behaviors according to the surrounding environment, without external intervention.

Modeling moral agent behavior is in line with the procedures being used to simulate human moral behavior. Although this simulation may provide a better understanding of human ethical choices and give a new perspective on moral in general, the lines under which an agent evolves its moral codes are yet to be set in order to be used as a mean towards building moral agents (Pereira and Saptawijaya, 2007).

2.4.4 Moral decisions

Considering the LoA used in the previous section, let us consider an Intensive Care Unity (ICU) with 2 (two) monitoring agents. Both agents interact with the environment reading the patients monitoring data, either it comes from cameras, oxygenation level reading devices or electrocardiograms. In a similar way, both agents can alter their procedures, such as altering oxygenation and temperature levels in the room or warning the medical team of the existence of any abnormality. They are also capable to predict future situations, extracting rules from past situations for future use, and to integrate them in the depths of their soul. These agents are ruled by a set of ethic norms, having as their ultimate objective the provision of the best possible service to the patient. Presenting a scenario in this way, are these agents moral agents? According to the LoA of moral agent, one may conclude that yes, they are. Both of them are moral agents, however, one is a human agent, while the other is a software one. In fact, both will be able and probably will commit os ethically dubious, if not incorrect; however, it is clear that contrary to the second case, the responsibility of their actions reflect only upon themselves. In other words, the responsibility in the case of the agent is not so clear to be defined concerning the entity it should reflect, if the agent itself, or its owner, or even other environment input. Its certain that from the developers LoA, an agent is not as independent, proactive, or interactive as it resembles, once he/she set the rules that the agent has to follow. However, taking into consideration its capacity of adaptation, it is expectable that in the short term it may remodel itself into a version completely distinct from the former one. It acts in the same way as a father that educates a son and transmits to him/her his moral code. However, there is always the question: to whom should be inputted the responsibility of their future actions.

Although the enlargement of the moral agent class in order to include the existence of virtual agents which are also moral agents, is not consensual, i.e. it is valid and advisable considering the inevitability of, during one of its learning cycles, a moral decision presents itself to the agent. It is thereby essential to define a set of principles that will allow an improvement of the agent development process, delimitating the frontier of action and principles that ensure, not only in the future, but as well as in the present, that these systems will work in synergies with society.

Although norms and regulations have been made for standardization of agent argumentation and communication, no similar approach has yet to be successful in the definition of properties that are essential for agents to have when taking actions in an areas such as medicine where sometimes little decision may have humungous ethical drawbacks. While a general ethics code was in fact developed by the Association for Computing Machinery (ACM), this code, though comprised of essential points which are essential for any area and technology, is by this reason not specific enough for the needs of agents developer in the healthcare area (ACM, 2009) A set of guidelines and rules must be defined to clearly state the characteristics an intelligent agent must have to be considered moral agent as well as the division of the developers responsibility, and the major role taken by the environment through machine learning techniques.

Chapter 3
Formalisation of the archetype

Introduction

Considering the complexity and heterogeneity inherent to the creation of an interoperability architecture and modelling of information flow, in this chapter it is proposed an archetype for healthcare interoperability based on multi-agent systems.

3.1 Archetype requirements and analysis

Firstly it is necessary to enumerate and detail the basic features of a interoperability platform in order to better understand the role and advantages of using multi-agent systems as a backbone of these platforms. Many of these properties are interrelated in the sense that they can be complementary to each other. Henceforth the main features are:

Ability to interact with information systems as loosely coupled services

There are conceptual and technical distinct standpoints when working in interoperability and integration of heterogeneous systems. When it comes to integration, it can range from loosely coupled to tightly coupled systems and further to fully integrated systems. The distinction among them relates to the degrees of liberty among each other (Vernadat, 2010). Full integration implies that the systems are no longer distinguishable from each other in the integrated systems as a whole. On the other hand tightly bounded systems although distinguishable are so intertwined and specific any modification or status change in one of the systems can have direct impact on the others. In loosely coupled integration the different system components are kept autonomous and still continue to exist and function on their own independently of other systems failure, while still being able to communicate and work as components of an integrated systems. Henceforth, loosely coupling of systems is often associated as a unequivocal characteristic that moves integration towards greater interoperability.

As it is self-evident from its definition, one of the main requirements of interoperability is easiness of addition and removal of involved entities. This property is henceforth of the foremost importance for an interoperability platform and implies that although exchanging information among themselves, all services must be able to function independently from each other. This provides stability to the overall architecture and systems are also less faulty to complications or alterations, as well as they provides the capability to remove, add or update any system in the existing infrastructure, something greatly limited to management decisions in fully integrated environments.

Adaptability to non-standard or flavored systems

Although the use of standards of communication is aimed towards the looseness of systems within heterogeneous environments, the lack of higher levels of semantic or ontology based understanding results in different implementations and understandings of the same standard. Similarly to the reverse of a coin, the ability to a standard to adapt to a wide variety of information and knowledge that might occur in one environment, also makes these standard open to interpretation and flavoring when being implemented.

Independence from legacy systems

Within any HIS there is a considerable dynamism within the information infrastructure as new systems are added or existing are either removed or updated. This may be partially more evident due to the association of the continuous evolution of medical science and information systems as well as to the necessity of IT management to search for more convenient providers as they appear or maintenance contracts are renegotiated.

The ability to add or remove these systems does not only requires a loosely coupled but also the existence of a consolidated information data source where all systems information is integrated. Without this consolidation of information when a system is removed from functioning there is often the need to maintain a legacy connection for the past information. These legacy systems can be costly financially and also in terms of the performance of the overall infrastructure.

Consolidation of information

Interoperability among heterogeneous systems with highly complex workflows and information flow can result in incoherences among these different systems. The consolidation of the overall information within the HIS in a structured and centralized manner can address issues such as the functions of a master datasource in case of inconsistencies.

Moreover, the existence of such data structure permits that once a systems is down due to failure the HIS infrastructure status continues on. The meaningful information of one system that performs operations towards interoperability should be made available even when it is down. However, for that purpose, all that information from distinct systems needs to be stored and consolidated from their specific data-models to another ones. This creates a trade-off. A data model needs to be more generalist in order to store all information which may be necessary to the HIS.

This consolidated database of information can not be a point of failure for the whole platform infra-structure. So it has to be redundant, but also the platform must be able to perform its functions as intended without the database.

Intelligence

The healthcare environment, due to the high complexity of the heterogeneous systems involved requires an adaptive and autonomous set of behaviors that prevents complications in the interoperability process within the HIS. Those behaviors also react autonomously and can correct existing failure situations.

Moreover, higher levels of communication such as stated by ontologies or involving semantic reasoning are important for healthcare information due to the necessity to address issues of distinct terminologies, which are not shared among all the information systems in the HIS. Intelligent behaviours are essential to predict and validate, both semantically and syntactically, the data and information gathered according to the knowledge representation and reasoning techniques used in each middleware service responsible for systems interoperability.

Embedded load balancing end redundancy of services

Any interoperability platform with the mentioned characteristics is bound to have a need for high processing capabilities and resistance to failure of distinct components of its own architecture. For this purpose the services provided by this platform must be distributed as well as redundant. The scalability and modularity of this architecture is essential not only to the selection of new information systems but also for the development and implementation of Decision Support Systems (DSS). The multitude and intricacy of services that must be performed by DSS or Group Decision Support (GDSS), require such a characteristics or otherwise would be inefficiently intertwined with other essential solutions such as the Electronic Health Record (EHR).

3.2 Archetype technical architecture

There are numerous methodologies to implement both multi-agent systems and interoperability. In the proposed archetype several modules are interconnected with different and clear dependencies in order to grant it scalability and ease of implementation. Although dependencies among modules are unavoidable, they can result in a monolithic architecture when one considers practical applications. Henceforth, the methodology proposed divides the application development in the modular cores, which are related with the perspective of software development:

- Core Agent Framework
- Database Independent Persistence Module
- Web-services Tier
- Ontology and Terminology Server
- Agent Development Tier

3.2.1 Core Agent Framework

There are numerous frameworks available, which provide a core environment for the development of agent based systems.

The main properties aimed in such a core module are:

- extensibility
- reliability
- continuous development
- FIPA compliant

The select framework for the development of this project was the JADE and WADE framwework.JADE (Java Agent DEvelopment Framework) is a software framework natively developed in the Java language. JADE is responsible for the core behaviour of the agents, defining its internal core processing and reasoning. It is software development framework aimed at developing multi-agent systems and applications conforming to FIPA standards for intelligent agents.

Henceforth the backbone of this framework contains two main features, a FIPA-compliant agent platform and a package to develop Java agents (Bellifemine et al, 2010a).

JADE is composed of the following main packages (Bellifemine et al, 2010a,b, 2007):

- ***jade.core*** this package implements the kernel of the system. The most visible and main concept of this framework is the Agent class that must be extended by application developers. Another key concept is the class Behaviour and the other classes that implement its hierarchy are contained in *jade.core.behaviours* sub-package. Behaviours are the core reasoning of the agent and implement the tasks, or intentions, of an agent. In more practical terms they are units of code that contain the agent's logical activity and that can be composed in various ways to achieve complex execution patterns and that, though being usually characterized by a single running thread, can be concurrently executed. Application programmers define an agent's internal reasoning by writing behaviours and agent execution paths interconnecting them.
- ***jade.lang.acl*** this sub-package is one of the core tools provided by JADE and provide access to process Agent Communication Language according to FIPA standard specifications
- ***jade.content*** this package contains the set of classes that allow and aid in the development of user-defined ontologies and content-languages. These encoding tools allow to send serialized user build ontologies as content elements of an ACL message. The default codec for this process is contained int he class *jade.content.lang.sl*, which contains the both the parser and the encoder for the SL codec.
- ***jade.domain*** contains all those Java classes that represent the Agent Management entities defined by the FIPA standard, in particular the AMS and DF agents, that provide life-cycle, white and yellow page services. The subpackage *jade.domain.FIPAAgentManagement* contains the FIPA-Agent-Management Ontology and all the classes representing its concepts. The subpackage *jade.domain.JADEAgentManagement* contains the JADE extensions for AgentManagement, including the Ontology and all the classes representing its concepts. The subpackage *jade.domain.introspection* contains the concepts used for the domain of discourse between the JADE tools and the JADE kernel. The subpackage *jade.domain.mobility* contains all concepts used to communicate about mobility.
- ***jade.gui*** this package contains a set of generic classes useful to create GUIs to display and edit agents, ACL messages and other objects otherwise only available programmatically.
- ***jade.mtp*** contains a Java interface that every Message Transport Protocol should implement in order to be readily integrated with the JADE framework, and the implementation of a set of these protocols.
- ***jade.proto*** contains classes to model standard interaction protocols (i.e. fipa-request, fipa-query, fipa-contract-net, fipa-subscribe and soon others defined by FIPA), as well as classes

to help application programmers to create protocols of their own. The FIPA package contains the IDL module defined by FIPA for IIOP-based message transport.

- ***jade.wrapper*** provides wrappers of the JADE higher-level functionalities that allows the usage of JADE as a library, where external Java applications launch JADE agents and agent container

This module using JADE as the base agent development library, extends it adding capabilities to the agent which, were previouly unavailable. Henceforth, this is not merely a module of reference for the JADE Framework, but rather an extension of several capabilities. Among these new features emphasis should be placed on:

- continuous monitorisation of activity and individual agent statistics
- embedded hibernate features
- persistent arguments and variables
- environment configuration automation
- embedded internal persistent log monitor

Most of these features were achieved thought the extension of the jade.core.Agent class by a aida.core.Agent class which implemented several methods and used different modules.

3.2.1.1 Continuous Agent Monitorisation

The continuous monitoring of activity enables statistics regarding the process time occupied by each agent in every container and platform in the CPU

3.2.1.2 Embedded Hibernate Features

The utilisation of Hibernate implicates a creation of a Session Factory, which is responsible for the creation of a database connection in a safe and easy manner. From the perspective of the Agent Paradigm each agent is an individual, with independent life-cycle and behaviour, for this reason each agent must be capable to manage its own connections according to its own functioning. As this object is thread safe one session factory for each datasource and agent and should be shared among all behaviours of the agent. This implicated internal hibernate features in the Agent class, creating a mapping for each datasource available and the corresponding created Session Factory through the instance of the Session Factory Store object within the Agent class.

Henceforth within each behaviour which a pointer to the agent which instantiated it can call a method that returns in a thread safe manner an open connection to the needed datasource. Several simultaneous connection can be established and maintained for the same or distinct datasources by distinct threads without concurrency problems.

3.2.1.3 Persistent dynamic arguments

This feature provides an active set of dynamic arguments which can persist through all states of and agent life cycle. These arguments are loaded at the creation of the agent and keep synchronous throughout its inner instructions. The persistence of these variables is extremely important for handling agent recovery to failure and to easy and dynamic agent arguments at boot level.

These arguments are kept in a XML file stored in database, which is loaded and persisted through the inner behaviours of the agent.

3.2.1.4 Platform configuration automation

When within an specific infrastructure, configurations such as firewalls or proxies are important to configure in every agent and inherent thread. Henceforth another extension of the Agent class enabled the definition of

3.2.1.5 Inner persistent log monitor

The persistent log monitor uses the imbued database persistent module to store internal log events directly into the database. As is visible from the Listing A.4, this internal process it stores unequivocally and agents log messages according to their type and date. Henceforth fine log detail handling can be managed within the framework structure.

3.2.2 Database Interoperation Interface

In regular information systems, the essential information generated needs to be kept and shared. Considering this system is aimed towards integration and dissemination of knowledge among heterogeneous systems, the persistence of information is henceforth a requirement. More than a single source of data, agents within this architecture must be able to access in a transparent and simultaneous manner distinct datasources.

There are several approaches to achieve persistence in a relational database, direct proprietary diver usage and connect approach through JDBC API, Enterprise Java Beans and a Object-Relational Mapping framework such as Hibernate, to name some of the most common. Each one provides strong and weak points, which must be evaluated when selecting the approach best suited for agent based interoperability. Considering the required characteristics for such a tier, the most significant are:

- database engine independence

- simplification for persistence of object oriented data structures in the relational model
- consistent view and access of data structures, queries and operations for all agents
- functional optimisation of queries

In order to avoid the use of pure SQL statements and falling in the need for compliant SQL with a specific database provider, EJB or a ORM are required for they dynamically create statements, queries and methods to handle with the specificities of each database engine.

Moreover, according to what as was previously mentioned, the intended aim of this tier was to gather data modelling and persistence functions, making this business logic shared among all agents in the platform, and henceforth prevent inconsistencies among agents when exchanging information. This need to provide a consistent and shared data modelling between the relational and the object oriented paradigm points towards the usage of a ORM and of simple POJOS, which can be exchanged among agents and extended as simple BeanOntologies.

There are several ORM frameworks available for Java. Several were analysed and evaluated in order to select one to be naturally integrated int he platform. To name some of the most significant:

- Athena Framework
- Carbonado
- Cayenne
- Ebean
- EclipseLink
- Hibernate
- iBatis
- Torque

All of them present an approach towards the addressing of the impedance mismatch conflict and methodologies for ease in developing database interfaces for applications. However, as superficially described in the Section this platform contains an hibernate oriented extension embedded in the core Agent class itself as persistence is considered a requirement for all agents. At the time of the implementation of the archetype the hibernate framework offered a most stable and adaptable behaviour than others. Later on the JADE framework added its own integration with the Hibernate framework, however as it didn't fully respond to the needs that were already implemented in this framework it was not adopted.

3.2.2.1 Hibernate Description

Hibernate is considered an Object-Relational Mapping (ORM) library for Java, providing a framework for mapping an object-oriented domain model to a traditional relational database. This ORM aims to solve limitations and mismatch for data persistence shared among the object-oriented model and the relational databases model. The approach used with this framework was to use high-level object abstractions to handle direct persistent database interactions, which are flavoured and dependent to specific proprietary drivers. This high-level objects handle database sessions, connections and operations, such as queries, inserts and updates. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

In order to address the limitations of vendor specificities when it come to SQL statements, hibernate provides not only specific objects that allows dynamic construction of search criteria but also Hibernate Query Language (HQL). HQL contrary to SQL offer two advantages is being both completely oriented towards the use of objects and also being completely vendor independent. The Hibernate framework deals with the transformation of criteria and HQL queries directly into the database vendor specific query type and syntax.

The parametrization of datasources is stored in in XML files under one of their specification, in this implementation the Hibernate Configuration DTD 3.0 As is visible from the details on the example in the Appendix A.1 This configuration file contains information pertaining to the datasource connection and relational mappings. The configuration is associated to the creation of a session factory that manages creation of new sessions to the database specified. For each database, user and specification a distinct file must be used to create the session factory.

Where the database connection is concerned this configuration file, aside from the connection string, user and password that are usually required, defines a database dialect from the ones made available by Hibernate and a driver class. This information is important not only to locate the specific driver for the database but the language that will translate the general HQL statements to SQL but also to know which driver to use from the ones made available in the application *classpath*. Moreoever, there are other significant parameter relating to the connection that are defined in this file, such as:

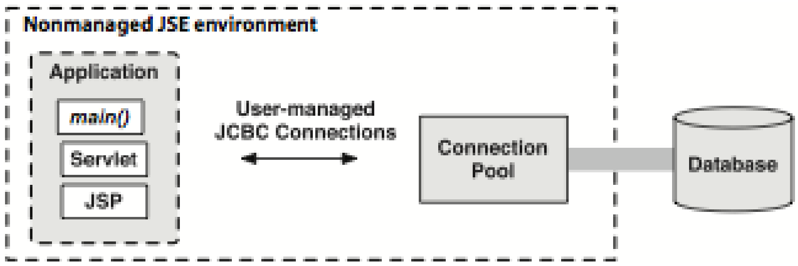
- auto-commit status - defines if each SQL statement is committed directly in the database or if a commit must be made in the command so that the changes are persisted.
- jdbc batch size - size of the list of SQL commands that are stored in cache before being executed in the database.
- session context - defined the context level the session factory will bind the session to. The default is thread, henceforth the session will be associated to the thread that called the method openSession().
- cache provider class - defined the class that will be used by hibernate for the second-level cache.
- show SQL - interesting feature for debug purposes.

A more complete number of the most significant parameters can be found at Table A.1.

The session context of a session factory is a determinant and useful method in this specific implementation with agents. Considering each agent is a thread, ergo possessing a dedicated line of process only, the sessions of each session factory that is allocated to an agent is only accessible by itself, following the paradigm of agent independence and autonomy.

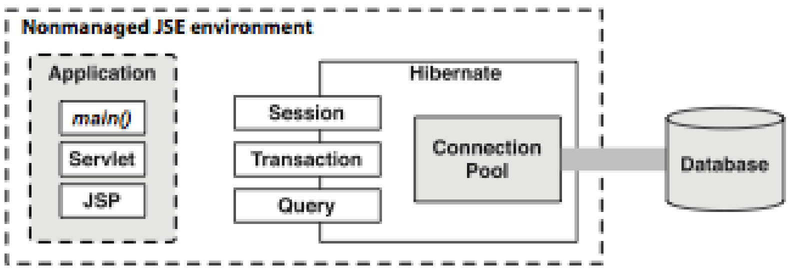
Another important feature of hibernate is the association of connection pools and two distinct levels of cache that aim to improve, but in an adaptable manner, the response of the application against the database.

Fig. 3.1: JDBC connection pooling in a non-managed environment (Bauer and King, 2006).



Without Hibernate, as is visible in Figure 3.1 the application calls the connection pool to obtain a JDBC connection and then executes SQL statements with the JDBC programming interface. When the application closes the SQL statements and finally closes the connection, the prepared statements and connection aren't destroyed, but are returned to the pool (Bauer and King, 2006). Under his architecture the sessions and statements are directly managed by the application code, which presents many difficulties in complex software development.

Fig. 3.2: Hibernate with a connection pool in a nonmanaged environment (Bauer and King, 2006).



The use of Hibernate changes this architecture into a more abstract one as it is demonstrated in Figure 3.2. Hibernate functions as a middleware in all interactions of the application with the database JDBC connection pool. Hibernate defines a plug-in architecture that allows integration with any connection-pooling software such as C3P0. C3P0 is an easy-to-use library for augmenting traditional (DriverManager-based) JDBC drivers with JNDI-bindable

DataSources, including DataSources that implement Connection and Statement Pooling, as described by the jdbc3 spec and jdbc2 std extension.

Not only the existence of connection pools is important for decreasing waiting times for database operations. There are several resources that can be saved by managing and caching the access to common instances of objects. The level 1 (L1) cache ensures that within the same session requests for a given object from a database, it will always return the same object instance. Henceforth it prevents the retrieved data from establishing conflicting instances. Moreover it prevents Hibernate from trying to load an object multiple times.Items in the L1 cache can be individually discarded by invoking the evict() method on the session for the object that you wish to discard. To discard all items in the L1 cache, invoke the clear() method (Minter and Linwood, 2006).

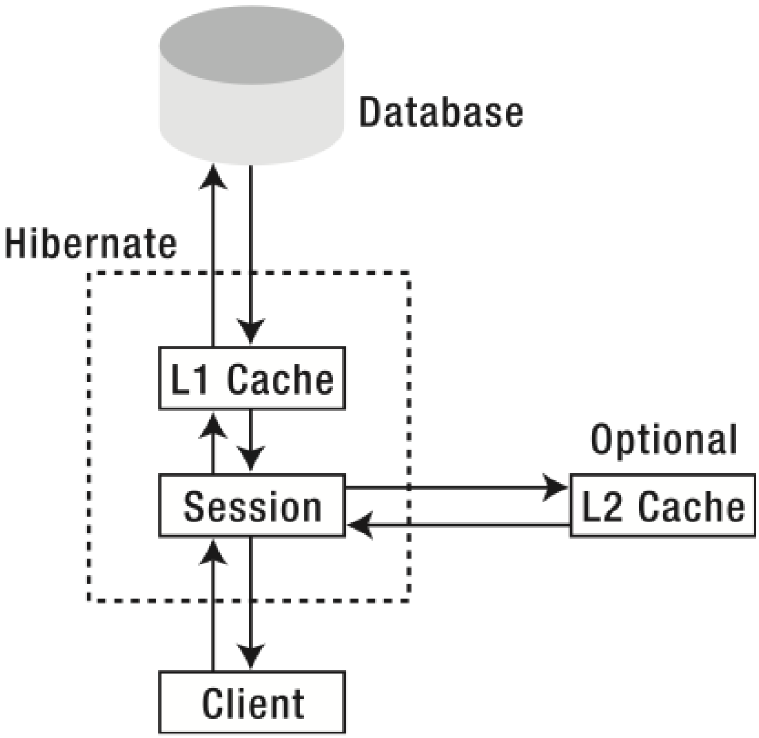
The second level of cache is different from the mandatory L1 in the sense that the Hibernate based application gains the advantages of a client-side database cache with with properties that exceed the one available by L1. It configures a cluster or JVM-level (SessionFactory-level) cache on a class-by-class and collection-by-collection basis.The second-level cache holds on to the information for all properties and associations and collections for individual entities that are marked to be cached.

There are several implementations of second level cache than can be added into an Hibernate configuration. The Table A.2 gives a list of the most common providers that are possible to be added into Hibernate.

Through the combination of both the levels of cache, Hibernate has a major advantage over the traditional connected approach. Figure 3.3 shows the two caches available to the session. The compulsory L1 cache, through which all requests must pass, and the optional level 2 (L2) cache. The L1 cache will always be consulted before any attempt is made to locate an object in the L2 cache. It is visible that the L2 cache is not integrant to to Hibernate. Although it can be accessed via the session in a manner that is transparent, this L2 cache is a pluggable interface to any one of a variety of caches that are maintained on the same JVM as your Hibernate application, or on an external JVM. This allows a cache to be shared between applications on the same machine or even between multiple applications on multiple machines (Minter and Linwood, 2005), which is an important feature for a distributed application as a multi-agent system.

The configuration file also contains a list of Hibernate Mappings (HBM) configuration files that are mappings between the objects and the database schema. The mapping document is designed to be readable and editable. The mapping language is oriented towards java development, which implies that the mappings are constructed regarding persistent class declarations and not table declarations. In another words, the xml file that is referenced in the session factory configuration contains several attributes that map the java object that Hibernate handles

Fig. 3.3: The sessions relationship to the caches (Minter and Linwood, 2005).



and the database schema that this object represents. A example os such a mapping that was implemented in this archetype is provided in the Listing A.2 and Listing A.2.

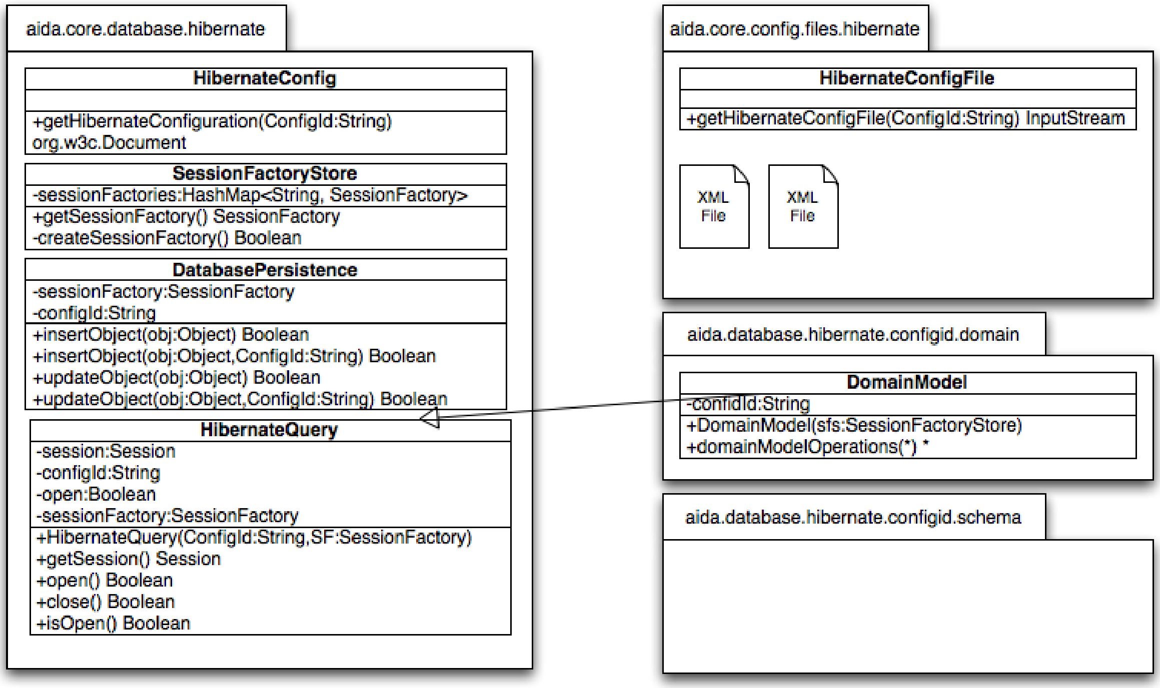
In the provided HBM the class named *aida.database.hibernate.pce.schema.Pcedoentes* is mapped with the table PCEDOENTES. For each column in the database there is a mapping to the corresponding java variable name and data type. In order to avoid the use of markup language the name of the variable is significant to the mapping as it will be used by Hibernate for the usage of the getters and setters. There are constraints that can be added so that Hibernate is able validate the data contained within the java objected before preparing the database statements.

The latest implementation of this archetype includes the usage of both levels of caching with C3P0 and JBoss Cache 2. This offers several advantages for the agent development process.

3.2.2.2 Interface Implementation

As mentioned previously each agent possesses an internal set of thread-safe connection factories which are stored in a internal object of the agent, namely an instantiantion of the SessionFactoryStore class. This class, as it is demonstrated on Figure 3.4 contains a HashMap of SessionFactories and their unique configuration identification. The configuration identification

Fig. 3.4: Description of the database interfaces.



addresses one and only one SessionFactory and is ultimately the name of the hibernate configuration file present at the *aida.core.database.hibernate* package and made available by the HibernateConfigFile class at that same package.

3.2.3 Web-services Tier

Although agents can provide web-services quite easily, this option was considered as far too complex regarding the mobility and lifecycle the agent system must provide. Henceforth a web-service layer was independently created, so that the availability of this services could be ensured and improved.

However, though they not provider web-services directly they are still web-service consumers. This consumer tier is made available for the agents to store all internal and also external web-service clients and configurations in a centralised fashion to all agents. Henceforth there this tier is abstractly composed by a consolidation of all web-service client in a module of the agent platform and a Tomcat implementation of some necessary interoperability methods necessary to be shared with external systems.

3.2.4 Ontology and Terminology Server

This tier grants a agglomerated access to all needed ontologies and terminologies by the platform agents and external service providers. Within this centralised storage there are contained several healthcare standards and proprietary terminologies. Among the standard the most significant for interoperability is HL7 version 2 as it is in the field the most commonly used nowadays. The main aim of an interoperability archetype is to model and enable communication. Within this platform communication occurs among agents and non agent-based services. Default communication among agents is provided by its JADE core and follows all FIPA communications standards. In addiction in JADE there are numerous methodologies to implement agent ontologies through the usage of existing terminologies.

On the other hand communication between agent and a and non agent-based services can be implemented in this archetype in two main manners, pure ASCII over TCP socket or web-services. The most common HL7 servers and clients use a plain ASCII streams over TCP socket as a means towards communication between systems. However, as a standard, HL7 messages can be defined through an XSD schema and mapped as input and output for web-services or more complex services.

3.2.5 Agent Development Tier

The development tier is the backbone for the development of agents. It is the layer in which the agents logic and behaviours are placed upon. As it is understandable it stands on top of all previous layers being able to access them all transparently and inheriting their included methods and classes. Due to this top role it depends directly or indirectly from all others and gathers the specific agent development. As it stands on top of all other tiers it is obvious it allows that all agents are provided with the same resources and basic characteristics. In different words this tier encompasses the individual agent types and embedded behaviours of each agent, while maintaining the internal general behaviours.

3.3 Archetype conceptual definition

In order to address the characteristics that were detailed in the previous section there can be numerous approaches and technologies involved. However this archetype defends that agent-based systems address many of them naturally as well as present concrete advantages through its simplification in introducing intelligent behaviours into these systems. Bearing in mind the extreme complexity and heterogeneity that configures the development and implementation of interoperability in the scale of a HIS, modelling its information flow, in this section it is proposed an archetype for healthcare interoperability based on multi-agent systems.

Multi-agent systems are an interesting technology in the area of interoperability in healthcare. Being multi-agent architectures a field of research of distributed artificial intelligence, this technology is intrinsically connected to the basilar concepts that define a distributed architecture, while being distinct in the intrinsic definition of an agent versus the properties of the general middle-wares of many others distributed architectures. Being distributed by nature its characteristics introduce MAS as a rich and fiercely adaptable technology with great interest mainly due to the research interests in this arena. In this research area concepts and technologies such as terminologies, ontologies, mobility, failure recovery and intelligent behaviours are embedded or explored in many existing frameworks. Henceforth they of interest for healthcare interoperability and a tool towards intelligent interoperability systems.

The general abstract tiers of this interoperability archetype are:

- External connector tier (ECT)
- Internal interoperability tier (IIT)
- Web-services Interface (WI)
- Consolidation Database (CD)

One may consider that at the core of this archetype its the consolidation database, in which all information relevant for interoperability and clinical decision is validated and consolidated. Only the internal interoperability tier and the web-service interface have direct interaction with this information tier.

While the external connector tier is extremely adaptable in order to imbed the specific features of each of the services that functions within the HIS, the internal interoperability tier is oriented towards an internal ontology that includes not only agent events and task but as well exact mapping of the the data model existent in the consolidation database.

Due to the mobility inherent to the agent platform which is intended to be maintained and the static nature of web-services in production state, the existence of such a tier addresses the necessities of some systems to communicate via web-services. Where this service provided by an agent and being it mobile, there would be complications with the dynamic allocation of this service as the agent move though physical machines.

Currently there are numerous frameworks that provide a core environment for the development of agent based systems. The main properties aimed in such a core module are extensibility; reliability; continuous development; and FIPA compliancy. The proposed archetype was developed and implemented using JADE as the base agent development library, extending it and adding capabilities to the agent which, were previously unavailable, extending several of its capabilities. Among these new features emphasis should be placed on continuous monitoring of activity and individual agent statistics; embedded hibernate features; persistent arguments and variables; environment configuration automation; and embedded internal persistent log monitor. Most of these features were achieved thought the extension of the jade.core.Agent class by a *aida.core.Agent* class which implemented several methods and used different modules.

In order to further comprehend the functioning of this archetype its important to detail the interoperability process in one specific feature that associates several services within the his. For this purpose it will be detailed the process that manages the requests and executions of diagnose and therapeutic complementary exams, a the extremely heterogeneous and complex conjunction of services.

3.3.1 Notion of basilar services within the HIS

Within the healthcare environment there are numerous systems that need to interoperate among each other. However there are set of them that are horizontal to most department and can be considered the heart of the health information system. There systems are the patient management system, the radiological information system and the laboratory information system. Each one of them deals with core and important areas of the information and workflow within and healthcare unit.

The patient management system is responsible for managing the patient information, who he is, where he is, where he has been and such type of information. Usually all services use it to register the patient information and which medical acts have been performed on him for charging purposes. Regardless for this matter it should not be confounded with an electronic health record. The purpose of each one is distinct, as far from centering on clinical information and acts, this is system is more oriented towards management and registration of billing actions.

The radiological and laboratory information systems are very analogous in objective, each one is responsible for scheduling, reporting and storing information regarding the clinical exams that are associated with each one of them. While radiological informations systems deal with x-rays, computerized tomography or magnetic resonance, laboratory information systems deal with the execution of laboratory exams.

Although in each institution different information systems can be found these three core systems, though many implementations exist, their behaviour and functions are very standard. Henceforth, in the following subsection it will be modeled the core agent system and behaviours for interoperability with an patient management system.

3.3.2 Interoperability with the patient management system

The patient management system, as on of the core systems within an HIS and responsible for the management of all patients movements within the hospital, needs to be kept consistently with all related subjects in other IS. The optimal solution to this interoperability process would require that the PMS system would possess and SOA layer that communicated events from this system to another, or in this case this platform. The reality found in the healthcare environment in Portugal is different nonetheless and is based on the SONHO system, which for the meanwhile does not provide such feature. For this purpose in the basilar design of this archetype there is an agent that is responsible for all interactions with the PMS and solely with this system. The main function of this agent is to sync all changes within

Among the relevant information that needs to be kept coherently it is important to mention:

- patient personal data
- clinical episodes
- patient location
- inpatient transferences
- related death occurrence

The patient personal data consists consists in all information that relates to the patient identification, address, health insurance as is demonstrated by the following data model

This agent executes several independent processes recurrently, which can be represented by the following internal behaviours:

$$
\begin{aligned}
& Agt_{PMS} :: (importNewPatients, \\
& \quad actualiseUpdatedInformation, \\
& \quad createPatients, \\
& \quad mergePatients, \\
& \quad deletePatients, \\
& \quad fetchNewClinicalEpisodes, \\
& \quad createClinicalEpisodes, \\
& \quad fetchInpatientTransferences, \\
& \quad fectchDeathOccurrences, \\
& \quad);
\end{aligned}
$$

importNewPatients

This behaviour is responsible for introducing new patients that are added directly to the PMS in the entry of an inpatient or outpatient that hasn't previously been in the institution. This process is one of the few that is not dependent on events generated by triggers in the PMS. As new patients always possess an patient identification (pid) number that is sequential to the latest added patient, there is only a necessity to keep the highest pid in sync with the ones in the consolidation database.

createPatients

This procedure is used to create in the PMS patients that are existent in other information systems but are not present in the later. There are several reasons for this need, being among the most relevant sync of different PMS and resistance to failure of the overall architecture even while the PMS is non responsive. The need to sync among distinct PMS arrises from disperse information infrastructures where, due to some reason, two or more PMS function simultaneously in distinct but cooperative functional areas. This architecture require an asynchronous and automated creation of patients among the distinct systems.

Although synchronicity is an important characteristic searched for many purposes in services, due to the cornerstone value of the PMS, the implemented behaviour behaviour has shifted from this ideal. Considering it reacts to a list of creation events, it is oriented towards an on-request paradigm and its asynchronicity is intrinsic. This is understandable as means towards avoiding establishing the PMS as a point of failure to all other systems, and to automate the recovery procedures for failure events.

mergePatients

Due to human or system complications the existence of duplicate entries for the same patient may occur, being it either detected in the PMS or by external IS. When such a incoherency is detected an event for patient merge occurs and the PMS agent not only issues the patient merge in the PMS as disseminates this information to those services which are parametrised to receive it.

deletePatients

This behaviour processes a deletion of a patient event from the PMS and renders it inactive in the platform. Its occurrence is also propagated to other IS due to its significance, however the procedures associated to this event depend from the receiver system.

fetchNewClinicalEpisodes

Considering the significance of the clinical episodes to staff members during the clinical procedures within the hospital and to all information systems that are associated to them, the availability of new episodes needs to be nearly instantaneous. Similarly to the sync of new patients this behaviour searchers to gather the highest value of clinical episodes for each of the considered modalities.

createClinicalEpisodes

Similarly to the creation of patients the creation of clinical episodes is an asynchronous behaviour that is important for the functioning of other information services when then PMS is down. For this purpose there is also an platform mapping to clinical episodes so that other systems are able to function even when there is no clinical episode created in the core PMS.

fetchInpatientTransferences

This behaviour syncs the registry of inpatient transferences between services. All patient movements from beds and functional units are registered in the PMS and imported via this method.

fectchDeathOccurrences

This behaviour imports the registry of any patient death that occurs in the hospital or arrives at the patient morgue.

3.4 Model implementation - CICA

3.4.1 Conceptual environment of CICA

The Integrated Center for Ambulatory Surgery (CICA from Centro Integrado de Cirurgia de Ambulatório) was an implementation project of relevance that occurred in the *Centro Hospitalar do Porto*. This center although is part of the hospital center it has a certain degree of independence that extends to the information systems used. It has its own software for interaction with the patient'd health record and its own interface to handle requests and surgery reports. However, this system could not be an isolated island within the hospital. Henceforth, a interoperability process was required in order for information to flow between systems within the HIS. This new system needed to inform the other systems of the information it was storing and the exam requests to other services of the same institution. In order not to be an incomplete system it was also required to receive information on the patient and the exams he had scheduled at the hospital.

The conceptual interaction were similar to business to business process, in which two independent entities exchange information on request and necessity. However do the bindings between these institutions the volume and type of interactions were complex, as many dependencies where found in legal terms between these two entities.

The previously proposed archetype and core system were used in order to implement this interoperability process. Services provided by agents dispersed in different server for production and development environments were implemented in this project.

3.4.2 Business model

The foremost concern of this interoperability process is to make all distinct information systems within this physically separated institutions function synergistically towards the goals of the management that controls both. However it is important to bare in mind that some of the distinct core applications that function independently share the same functional purposes, but not the same responsibilities. In a more pragmatical perspective, the information systems of the main hospital center must keep all information not only for internal purposes but also of communication with the national healthcare service. The information regarding all that is financially supported by the Portuguese state and is performed in CICA must be stored in the SONHO information system or the state reimbursement wont be received by the hospital.

Although CICA uses an internal SAP system to manage billing of privately insured patients, all medical acts related to those patients who are supported by government based insurance, such as the basic national health system or others related to government workers, must be stored at the SONHO system. At the point of implementation the value of private insured patients was a minor part of the whole sum. This added an increased strain on the interoperability efforts to validate all incoming medical procedures executional information.

Another important set of information that is important to be kept in the hospital centre IS is the medical team that performed a specific surgery or other procedures. This is of the essence not only to be used for business intelligence purposes, but also given that all medical staff belongs and is paid by the hospital centre, this information need to be present in the human resources management IS that exists in the hospital.

Considering that the CICA, although with its degree of independence, is an integrant part of the hospital centre, it is important that the internal production of all exams produced are optimized synergistically. Henceforth, CICA must be able to request all exams that are performed within the hospital seemingly directly to the responsible information system as an advantage to increase internal service consumption.

Since all these requests go throughout the interoperability platform CICA requests this exams and procedures to an unique interface. This model allows consolidation of information that also of the essence to allow the extraction of production indicators to the head management of the hospital center.

3.4.3 Description of the key-role information technologies existent

Within CICA several information systems exist for similar tasks as the one provided by other systems within the central hospital HIS. Although independent in actions internally, most actions and information must be send from these systems to the HIS. All interactions from these systems with the AIDA platform are performed through an Oracle SOA system that manage all integration from the CICA's endpoint. Due to the need of shorter response times when issuing a request from the HCIS application, the electronic health record at CICA, all outgoing information from CICA is based on Web-Services (WS).

3.4.4 Information model

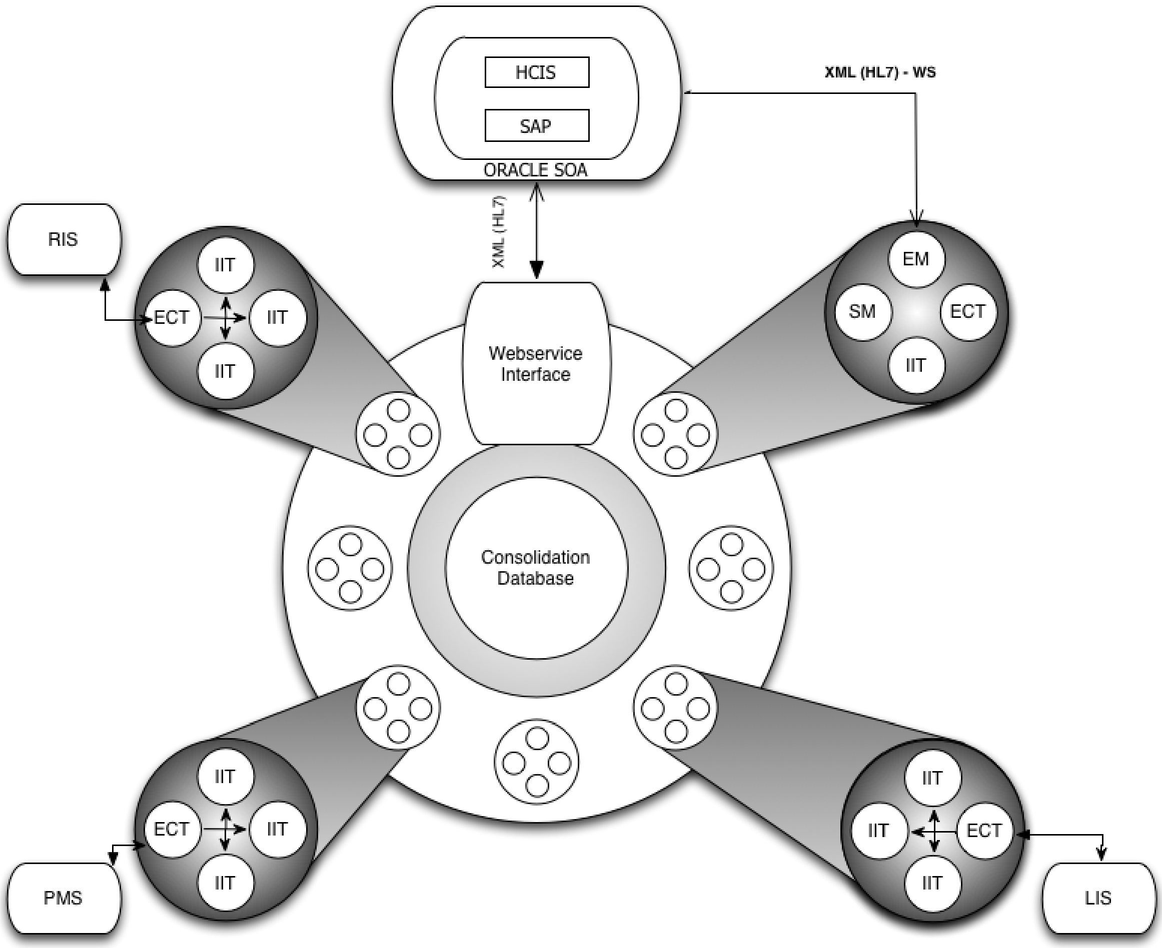
Most information exchanged from among these systems is based on an XML encoding of the HL7 version 2 standard. As it is understandable from this standard the segments and data can not be encoded directly as the norm states. The XML structure was not fully compliant with this standard as the encapsulating tags for the message segments and internal fields was in Spanish language.

The adaptability of the archetype to comprise middleware agents for such interoperability specificities is an important property in this case. It allowed these agents to be added into the architecture and be able to understand this information while transforming it into the archetype internal information data model and propagate it within the HIS.

There are different types of information that need to be exchanged among these systems. Depending on its purpose the information can be categorized into types as either production or clinical oriented. However, due to the dependencies of of both type, in most workflows for this model they are intertwined.

As mentioned earlier, to improve the internal consumption of resources, CICA is able to request exams seemingly directly to all other information services that produce diagnose and therapeutic complementary exams. In order to better understand the flow of information there

Fig. 3.5: Information flow for diagnose and therapeutic complementary exams requested form CICA.

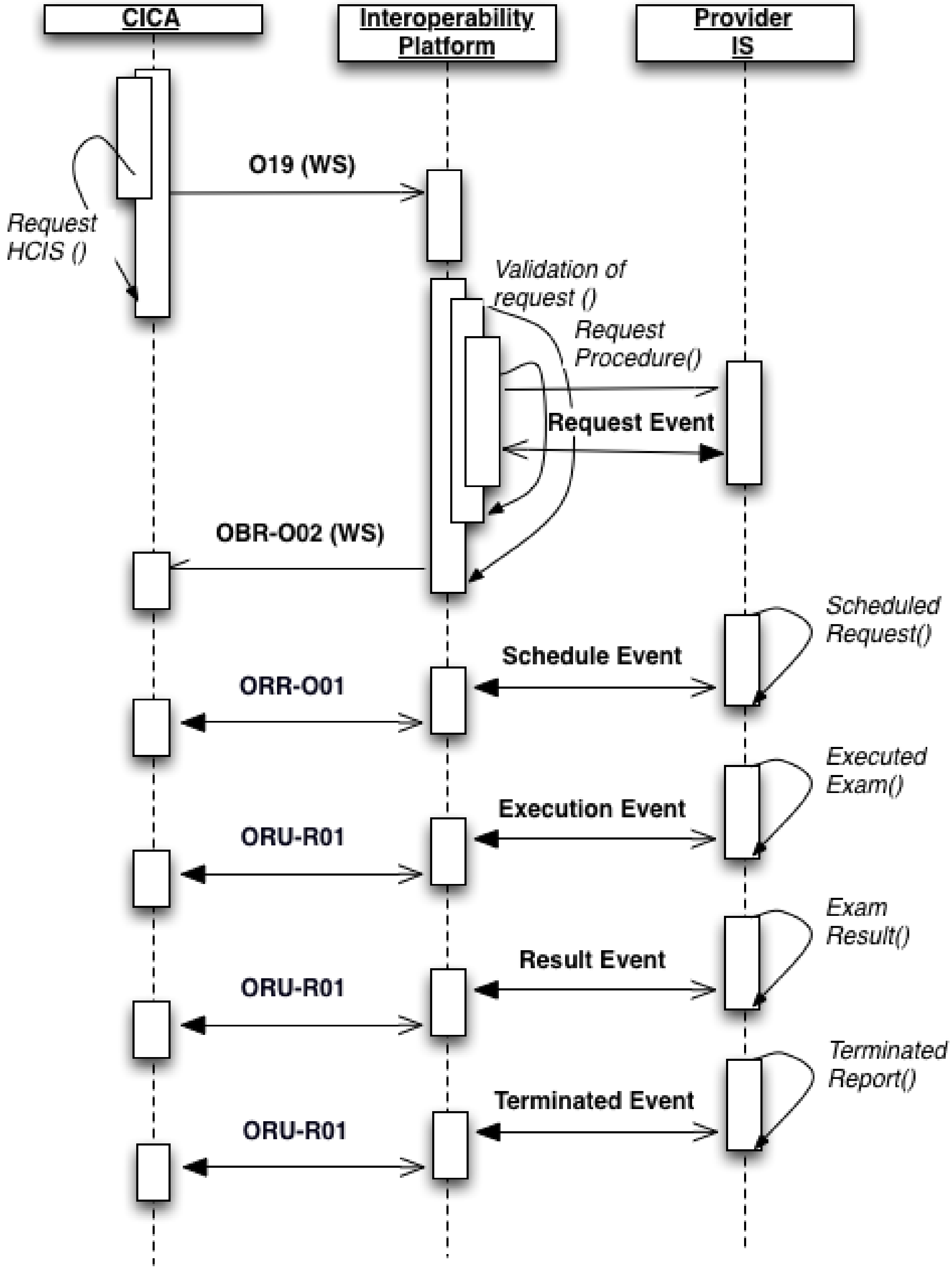


will be a generalization of the production tier as is visible in Figure 3.6. This last tier however can be a multitude of systems depending of the requested DTCE.

The request process, as is understandable in Figure 3.6, starts with a request in the clinical software at CICA. Through their SOA layer they send a request composed of different types of DTCE that are to be executed within the hospital center. An unidentifiable example of such a request can be found in Appendix A.5. This request is received through the WS layer and the request is integrated in a list kept in the consolidation database.

The list of requests is fetched by one instance of the Schedule Manager (SM) agent, which only processes specifically these events through filtering of sender, message type and event. In this case the sender is CICA the message type is an OMG, while the associated event is an O19. Due to the specificities of CICA's information model, this agent is only responses to the described requests. The process requires validation not the request, no only to general

Fig. 3.6: Information flow for diagnose and therapeutic complementary exams requested form CICA.



information but also the distinct DTCE that are in this message. The internal behaviour of the agents can be described by the following structure.

$$Agt_{SM} :: (fetchUnprocessedRequests,$$
$$processRequestArray()$$
$$);$$

fetchUnprocessedRequests

processRequestArray()

$$\$processRequestArray() := validateInformationConsistency,$$
$$changeRequestProcessingStateBusy$$
$$createHL7Message_{ORR-O02}$$
$$verifyPreviousRequest$$
$$processInnerPetitions()$$
$$sendHL7Message_{ORR-O02}$$
$$changeRequestProcessingStateResult$$
$$);$$

validateInformationConsistency

This method is responsible for validating the received request against the configuration of the institutions healthcare information system. There are multiple services within the the hospital center which from which DTCE can be requested from. This process validates if the requested exam is performed by the designed service and if there is any special workflow needed for the realization of that request. Special cases that can be mentioned are urgent requests, something that may arise during a surgery and exams which require superior permissions before being scheduled.

changeRequestProcessingStateBusy

This process blocks the request so that the request is only processed by one agent in concurrency prone environments. If unable to lock the state, the agents leaves the request and searches for another suitable request.

createHL7Message_ORR-O02

This methods creates the Hl7 messages ORR-O02 that is the response to a exam request. This is the main content of the message regarding the response to the message received, so that the CICA system knows which request this generated message answers to. The identification of the request is of the utmost significance considering the asynchrony nature of the communication process.

verifyPreviousRequest

Whenever possible this process aggregates requests in prior requests or actualizes the states of requests when its an update.

processInnerPetitions()

Most of the scheduling process and interoperation dependencies are at the inner petition level. A request main contain requests for a whole set of exams what may be performed by distinct services and departments, while having associated distinct cost aggregation objects.

sendHL7Message_ORR-O02

After being completely processed, the request has a corresponding HL7 answer to its contents. This message is of the type ORR-O02 and is sent to the SOA layer of cica services.

changeRequestProcessingStateResult

This process releases the status of the request by agent with several status codes that correspond to the answer by either the CICA SOA layer of the internal agent processing results.

As it was just mentioned previously most of the interoperability process is contained in the *processInnerPetitions* behaviour. It is composed by the following abstract methods:

$\$processInnerPetitions :=,$

$associateMedicalEpisode$

$sendSchedulingRequestORR - O02$

$structtureInnerHL7Response$

$);$

This process is responsible for managing the requests made by CICA and disperse them within the health information system. The *processInnerPetitions* function introduces the request into the consolidated knowledge repository, which henceforth generates events for the other information systems in order to proceed with the scheduling of the event. When the other systems generate an event regarding one of these systems it will be communicated to the SOA layer of CICA using the message type ORU-R01 as is visible in Figure 3.6.

As is visible from the Listing A.5 the HL7 message is not expressed in the regular standard, but rather encapsulated in a XML structure. For this purpose, a transformation of the standard was performed into and XSD document which was embedded in both the CICA SOA suit and the agents web-service layer.

3.4.5 Results

The implemented system entered in development since the mid of 2010, since then several increments and improvements have been performed until it entered in production in mid 2011. This platform is currently responsible for the interoperability of more than one service, however this study concerns only this specific set of interfaces and agents.

Although part of the same project and consolidated knowledge database, this system was set up on a distinct platform from the main interoperability main-frame in order to better test and evaluate the proposed archetype and the backbone framework created previously. The uniqueness of CICA's situation and configuration was used in order to attain a better degree of liberty during tests and developments. It is currently responsible for all the communication of this institution with the main hospital center.

The short term of development for a project of this scale indicates the agility and adaptation of the proposed archetype and developed tools. In addiction the current results point to a low time of response and lag reported by users and system failures have until now been avoided due to the replication of the agent platform in different environments.

The overall, results and feedback point towards a great adaptability of the archetype to the quirks and difficulties found in interoperability in healthcare environments.

Chapter 4

Results

4.1 Modelling intelligent behaviours in multi-agent based HL7 services

This research paper addresses the study of the devised platform and the improvements achieved by the addition of intelligent behaviours to agents handling communication with heterogeneous systems using the HL7 standard.

Using clustering algorithms these agents are able to detect when another systems is having communication issues and respond accordingly. The behaviour demonstrated by these agents is one of the core objectives mentioned in this thesis, embedding intelligent behaviours in order to improve healthcare interoperability.

IFIP Advances in Information and Communication Technology, Volume 335, Springer, 2010

Miguel Miranda, Gabriel Pontes, Pedro Gonçalves, Hugo Peixoto, Manuel Santos António Abelha and José Machado

Abstract

With the dissemination of Health Information Systems and the greater relevance of interoperability towards the quality of the information available to the clinical personnel, distinct architectures and methodologies have been devised in order to improve the existing platforms in the healthcare environment. However, most of them are based on HL7, an international standard for healthcare interoperability, which depending on the implementation as any technology has its advantages and limitations. This paper details the architecture and methodologies of a multi-agent based HL7 interoperation service. The mentioned system is incorporated in an integration platform, which is implemented in several healthcare institutions and uses different MAS to control and enable the flow of data and information within them. The log registry and extracted statistics of several years of interoperation in one institution are used to analyse the development of prediction models to imbue intelligent behaviour to the existing platform. The resulting models are studied and embedded into a validation HL7 server agent.

4.1.1 Introduction

The healthcare environment configures a paradigm of intricate information technology architectures, in which distinct solutions must share data and information. The exchange of data and information is of the essence towards the optimisation of existing resources and the improvement of the decision making process through the increase of the quality of information.

Numerous architectural solutions have been developed towards interoperability in healthcare, depending of the objectives, context and methodological approaches. At this architectural level, one can enumerate distinct and relevant abstract interoperability approaches, such as end-to-end, hub-and-spoke, distributed multi-agent or service oriented. Properties such as modularity, availability, scalability or delay timespan are associated to the interaction of different systems comprised in the devised architecture.

Within the healthcare environment the integration of all otherwise secluded applications is of the essence for the development of a scalable and functional Health Information System (HIS). A HIS can be defined as an abstract global information system for the processing of data, information and knowledge within the healthcare institution. It is therefore the consorted and integrated effort of the different heterogeneous solutions within the healthcare institution to collect, process, report and use information and knowledge relatated to its unique environment to influence the existing management policies, health programs, training, research and medical practice within this institution. (Kirsh, 2008)

Considering the definition of an HIS, its essence is the architectural model composed of a group of integrated and interoperable solutions within the healthcare institution. In contrast with the usage of a centralised solution, which is unthinkable considering the specificities of each areas of a healthcare unit, it aims to maintain all distinct services and solutions. It is henceforth essential to imbue the HIS architecture with the capacity to allow communication among different and otherwise secluded systems, avoiding their centralisation and dissemination of End-to-End connections, which restrict the growth of all the infrastructure associated to the HIS. The non-modularity of services adds complexity to alterations and improvements, increasing the global costs of the information systems.(Aier and Schönherr, 2006) Therefore, it is understandable the present concern demonstrated by distinct international institutions, responsible for financing and regulating the purchase and development projects for new HIS, with matters of flexibility, interoperation and integration of heterogeneous systems. (Berg, 2004) (PHII, 2004)

Congruently with these concerns, present tendencies regarding research and industry in interoperability applied to healthcare information systems, indicate the potential of agent oriented architecture. (Isern et al, 2010) Asides from modularity, scalability and adaptability these systems have also the potential to imbue new features associated to intelligent agents which may address the existing problems and solve important limitations otherwise difficult to tackle. Although healthcare standards like HL7 are completely distinct from agent communication standards, HL7 services can be also implemented under the agent paradigm. These agent based HL7 services can communicate with services that follow distinct paradigms and communicate with other agents using either HL7 or agent communication standards. Although the HL7 standard can be implemented using other architectures, agent based solutions enjoy of a vast interoperability capability, being capable to be embedded with the most particular behaviours. These behaviours can become increasingly effective if they use machine learning and other artificial intelligence techniques in order to adapt to the existing environment and being able to prevent and correct the flow of information and extraction of knowledge within the institution.

Henceforth, the beginning of this paper details the architecture and methodologies of a multi-agent based HL7 interoperation service. This system is part of in an integration platform, which implemented in several healthcare institutions and uses different MAS to control and enable the flow of data and information within them. The log registry and extracted statistics of several years of interoperation are used to analyse the development of prediction models to imbue intelligent behaviour to the existing platform.

4.1.2 Health Level Seven Protocol

Health Level Seven started as a mainly syntactic healthcare oriented communication protocol at the application layer, the seventh layer of the OSI communication model. This protocol defined the message structure to be exchanged by loosely connected healthcare applications by classifying the different types of messages involved in this environment with the aggregation of standardised segments.

The structuring and design of this standard, defining which artefacts of data should be transferred by a certain message, enabled and potentiated the application of HL7 in client-server architecture. (Ohe and Kaihara, 1996) The most common implementation of this architecture using HL7 is based on distinct socket communication clients and servers, in which the client sends an HL7 structured message to the server, that upon processing sends an acknowledgement HL7 standardised message. The HL7 standard is not bound to this architecture, but it is the most widely used in healthcare interoperability.

Although the initial standard was uniquely syntactic, the current version 3 is opening the HL7 scope towards semantic interoperability including the appropriate use of exchanged information in the sense of the communicating applications behaviour. The Message Development Framework (MDF) is currently moving towards the HL7 Development Framework (HDF), by that way shifting the HL7 paradigm from message to architecture. Newer HL7 developments such as the EHR-S Functional Model and the SOA Project Group activities have been pushing this move. (Lopez and Blobel, 2009)

4.1.3 AIDA Platform and HL7 Services

The AIDA (an Agency for the Integration, Diffusion and Archive of Information) platform was developed in order to support the diffusion and integration of information generated in the healthcare environment. This platform imbues many different integration paradigms, using mainly Service Oriented Architectures (SOA) and MAS to implement interoperation in a distributed, specific and standardised manner with all the service providers within an healthcare institution. Using this synergy it is maintained the independence and modularity of SOA and the intelligence and autonomy associated to MAS and Artificial Intelligence. (Machado et al, 2008) (Machado et al, 2007)

Being MAS a field of research in Distributed Artificial Intelligence, this technology is intrinsically relates with distributed problem solving, while being distinct in the intrinsic definition of an agent versus the properties of the general middlewares of the architecture called in its support. (Weiss, 1999) Indeed, under this approach a MAS subsumes a distributed architecture.

The MAS is able to manage through the agent life cycle the availability of the modules of the healthcare system and the HIS as a whole, while keeping all the agents that constitute the MAS freely distributed. In fact, new agents with the same characteristics and objectives can be created on-demand by the MAS, according to the necessities of the system they belong to. The structuring of these agents and of the MAS can be developed according to the services they provide and the logical functionality of the systems they interoperate with, allowing conceptually to take advantages of the SOA paradigm.

The core of SOA has as a key principle the division of large and complex problems into simpler and modular ones. However, contrary to the common methodologies to address a great number of complex problems, this architecture aims for the services of the smaller conceptual units to be achieved through complete independence. This methodology searches distinct areas of logic automation in order to unitarily make available a service that is part of a vaster and more complex service. It is ensured that in this way each of these units can be replaced by any other unit which performs the same service without concern. These base units can provide a service in a distributed way, independently and disassociated of the underlaying global services structure. (Erl, 2005) Although, at first glance, it may be extremely similar to the MAS paradigm, SOA is a concept not bound to one specific technology, it can be based on web-services, agents or any other technology following these basic rules.

Through the SOA paradigm, a system will not be dependent of its core units. It means that services can be easily replaced and updated, enabling modularity, scalability and independence. (Juric et al, 2007) These are the properties the AIDA platform associates the MAS and SOA paradigm in interoperability systems.

Henceforth, the agency's top layer of abstraction conceptually consists of 7 (seven) multi-agent based subsystems:

- AIDA-RIS - Radiological Information System;
- AIDA-MEIS - Medical Exams Information System;
- AIDA-LIS - Laboratories Information System;
- AIDA-ISM - Information System for Monitoring (e.g., vital signals monitoring);
- AIDA-PRM - Patient Relationship Management (including communication using SMS);
- AIDA-OWM - Organisation and Work Management (Including agenda, scheduling, planning and resource management; and
- AIDA-EHR- Electronic Health Records.

AIDA's devised architecture supports intelligent agents that acting as distributed entities on a healthcare environment, gather all the data, transform the underlying information, correct information incoherence and disseminate it thought-out the HIS. The introduction of data validation allows to improve the overall quality of information extracted and avoids the spread of inconsistencies over all involved systems. The quality of the gathered information is important to guarantee that the decisions made in these environments are based on sound principles and are not led astray by incoherent or inexistent information.

This platform was engineered under the perspective of a centralised repository for all the significant data in a healthcare institution. Under this happening, the underlying information must be retrieved from and disseminated towards the different service providers in the healthcare institution. Henceforth, the gathered information must be processed and corrected, validating the execution of existing workflows and ensuring the Quality-of-Information (QoI) disseminated to other systems. The presence of a global information and knowledge repository that is oriented towards the whole of the healthcare institution, that has the extensibility to adapt to the heterogeneity within its environment, is an important tool for information validation and knowledge discovery. For this purpose the AIDA platform uses an ORACLE RAC database, structuring most of its clinical and management information in highly compact but well structured XML syntax. The use of XML grants the necessary malleability to adapt the repository configuration to the needs of a specific institution, service or external provider. Clinical reports and other information gathered by the platform are therefore structured and processed in this repository and validated against existing knowledge or information.

Intelligent behaviours are essential to predict and validate, both semantically and syntactically, the data and information gathered according to the knowledge representation and reasoning techniques used in each middleware agent responsible for systems interoperability. Each of the agents is embedded with the explicit particular behaviours congruently with the particularities of the service provided in order to guarantee good quality of the information exchanged among any particular agent it is meant to interoperate with.

The scalability and modularity of this architecture is essential not only to the selection of new solutions but specially when developing Decision Support Systems (DSS). The multitude and intricacy of services that must be performed by DSS or Group Decision Support (GDSS), require such a platform or otherwise would be inefficiently intertwined with other essential solutions such as the Electronic Health Record (EHR) (Miranda et al, 2008) (Duarte et al, 2009).

Within its several MAS modules, the AIDA platform contains a proprietary communication system which implements service communication via HL7 standardised messages. This service is one of the core dissemination methods for medical information within the healthcare institutions and as mentioned before is oriented towards a service oriented paradigm. Each agent works either as client or server for a specific service within the healthcare institution. These agents reactively receive or send information for their specific system and have no other interaction with these external systems. Any of the described top layers of abstraction, can communicate with HL7 specialised client agents that exist within their MAS to exchange information with other MAS or external information systems.

4.1.4 HL7 Activity study

From the previous activity recorded by the AIDA's HL7 services, behavioural patterns of the integrated systems may be extracted in order to increase the reliability and performance of interoperation. The initial models are essentially directed towards the analysis of current load

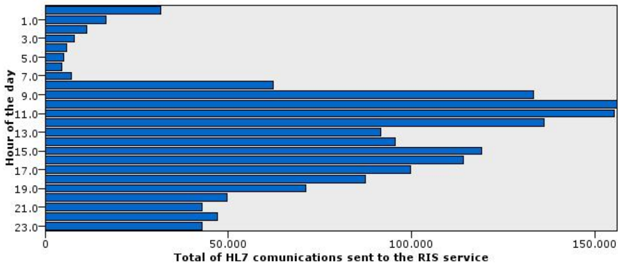


Fig. 4.1: Study of HL7 communications sent to the RIS service depending the time of the day

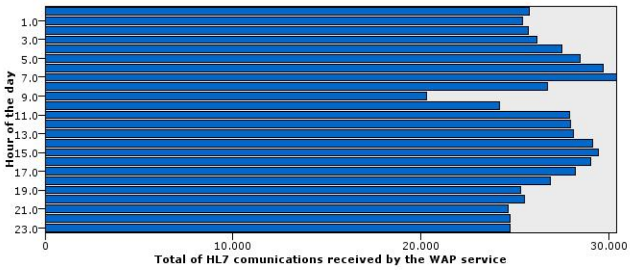


Fig. 4.2: Study of HL7 communications received by the WAP service depending the time of the day

over the existing architecture and prediction of existing bottlenecks. However, with the existing data further information can be extracted in order to develop models which can be embedded in order to overcome limitations and problems of these systems.

Bearing in mind the implemented architecture of HL7 and the underlying nature of service oriented paradigm, the study of such systems should be oriented towards a specific case and context. The model created for a specific service most surely will not express the functional behaviour of another, which inherent properties are usually distinct from each other. Besides, adaptive models must also be dynamically updatable and be able to learn changes in the environment they interact with. The influence of time of day over a service behaviour is understandable, as services which require human interaction are more active during regular work time due to scheduled external consultation, while automated services regarding monitoring are more regular throughout the day. As an example, in Figure 4.4 one can notice that the peak of the messages sent to the radiological information system (RIS) service coincides with the working day hours of the healthcare institution. On the other hand the automated chemical analysis service (WAP), as demonstrated in the Figure 4.2 has a continuous activity thought the day. Models directed towards the analysis and prediction of communication behaviour must adapt to this constraints in order to provide contextualized information and handle the regarded data.

Henceforth, the learning approach to interoperation agents can or can not be segmented into daily periods of time depending on the usage of the specific service, this segmentation however is of the essence for the proper classification and prediction of bottlenecks or systematic failure.

4.1.5 Modelling approaches

The selected approaches are dictated by the objectives and associated problems of the matter in study. One of the main concerns in the healthcare environment, combined with the need to increase services availability and reliability, is the instant response to errors and failure. One of the problems, which can be modelled, seems to be the regular need to reset HL7 server connections due to the loss of communication and incapability of these systems to detect and react to correct any faults.

To understand the implications and complexity of the detection of these events consider the case of the communication between a RIS and the medical emergency software. Regardless of the requesting method implemented, if the performed exam information is being disseminated through HL7, the failure to communicate the radiological complementary diagnosis method requested is hard to detect even with heuristic methods. Systems with no intelligence can not determine by themselves wether or not the fact that an exam for example is not available is due to a loss of systems communication. This will require additional effort by the physicians and technicians to access the exam stored at the PACS in the RIS as there is no information of its existence. In different words, it will be far more probable that the end user will be the first in detecting the failure in the information workflow via HL7. More than just radiological exams information is increasingly performed by HL7, medication, analysis and other services in the healthcare area are now integrated using HL7. In these cases more than revenue is at stake, the quality of the provided healthcare service may be harmed by this limitation.

To predict the loss of communication and errors in the HL7 server service it is proposed to analyse the time between HL7 messages of each service. Although this study is limited to the proposal of approaches and methodologies, it will be validated against existing data of one of the most active services in an healthcare institution, the RIS. Regarding the time of study, it will concern two years of interoperation communication via HL7. In light of the objective aimed with this analysis and the previous knowledge of each services specificity, data must be segmented into services and time of day.

Considering the volume and attributes to analyse, from the available techniques to mine the existing data and extract patterns that might enable the induction of anomalous situations, clustering seemed to provide the most adequate set of tools. The objective was henceforth to create clusters of timespans between messages, which might allow the detection of values

Table 4.1: K-Means Clustering Centres - RIS Timespan Between HL7 Messages - Study A

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
37,29	2766038,00	10103,55	64661,00	3254,04

Table 4.2: Number of Cases in each Cluster - Study A

Cluster 1	1590327
Cluster 2	1
Cluster 3	404
Cluster 4	1
Cluster 5	3391
Valid Cases	1594124

beyond the expected for a specific service on a give time. With this knowledge interoperation service agents could predict the loss of communication, warning system administrators and prevent the existence of incoherent information among supposedly integrated systems.

A first study (Study A) was performed using the overall clustering of timespan data with the K-Means algorithm and 5 centroids. The resulting clustering information displayed in Table 4.1 and Table 4.2 showed that over 99% of the cases were inside a single cluster with a rather low centroid. The maximum value within the mentioned Cluster 1 was of 1767 seconds, meaning that the maximum time between messages within this cluster was approximately under 30 minutes. The remaining cases could be considered as exceptional events with really high and high volume of them conciliated with the existing socket reset logs of the agent system. This indicates that this cases should be managed by the agent system as system failure and loss of communication.

However, considering the functional activity of the service being analysed, it is inferable by the study of the existing data that most of the activity is found within a limited daily time-frame. A comparative clustering study (Study B and Study C) using segmented data further displays this effect by the difference found in the first and lowest timespan between messages centroid. Over 86% of the overall activity in this service was performed during the time frame of Study B, as on the Study C the activity of this service is limited to emergency or internment occurrences. This causes that the first centroid has far lower value in the Study B than in the Study C. Withdrawing conclusion over the values of the other centroids is rather hypothetical, however they are connected with the need of server reset or other complications to the flow of information.

The one simultaneous fact considering the performance of an HL7 server and the workflow of an healthcare service is the existence of a distinct cluster of regular expected values that

Table 4.3: K-Means Clustering Centres - RIS Timespan Between HL7 Messages Between 8 A.M. and 8 P.M - Study B

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
24,75	15382	620,29	2766038	64661

Table 4.4: Number of Cases in each Cluster - Study B

Cluster 1	1360928
Cluster 2	2
Cluster 3	10297
Cluster 4	1
Cluster 5	1
Valid Cases	1371229

Table 4.5: K-Means Clustering Centres - RIS Timespan Between HL7 Messages Between 21 P.M. and 7 A.M - Study C

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
82.40	2489.31	30637	6788.74	12779.25

Table 4.6: Number of Cases in each Cluster - Study C

Cluster 1	218164
Cluster 2	3879
Cluster 3	3
Cluster 4	711
Cluster 5	138
Valid Cases	222895

depends on the regularity of the flow of information. The cases within this cluster are with an high degree of certainty examples of regular performance within the service. Henceforth as the time between HL7 messages falls outside the limits of this cluster it indicates the existence of an error with the flow of information within this system. Further studies variating the number of clusters would still create a cluster with impressively higher concentration of cases and lower timespan.

4.1.6 Embedding adaptive learning behaviours

With the resulting model indicating that regular cases within the system interoperation process usually create a stable concentrated cluster, such concept can be embedded into the multi-agent system in order to evaluate the current behaviour of the service. On regular distant intervals the agent evaluates the distribution of the information flow determining wether the service has lower activity intervals. The segmentation of these intervals as demonstrated before is important to add more sensibility and accurate knowledge to the agent. Depending on this analysis it segments or not the data into different datasets running the clustering algorithm and extracting the centroids and boundaries expected to each cluster in each segmented dataset.

The HL7 server agents were imbedded within this knowledge and as the timespan between messages approximates the upper boundary of the lowest centroid it generates warnings to the system administrators. However, when the server agent detects that it has left the upper boundary for this cluster it runs incrementally networking and thread reset/cleaning procedures.

4.1.7 Conclusion

With the massive introduction of information systems within healthcare institutions the relevance of the quality of the information they provide and the reliability in their performance became an essential requirement. As secluded systems ultimately work as hidden repositories of information the process of interoperation between all systems is an important directive in HIS management policies. However, as interoperation becomes a regular process the breakdown of these processes greatly diminishes the quality of information available. For this reason, embedding intelligent behaviours in order to enable to predict, prevent and correct such complications is an opportunity to both study the boundaries of intelligent agents and to improve the quality of service within healthcare interoperability.

The study of past interoperability processes indicates clustering the corresponding data, when the system is in full working production environment, a main cluster with the regular time between information exchange. This fact is understandable considering that a stable and working HL7 server will process this messages in regular intervals and will rarely loose socket connections, although that will eventually happen resulting in abnormal timespans between messages until the end user or system administrator notice and correct this factor.

This model that was embedded into the agent based HL7 server is a simple example how unsupervised learning capabilities when properly explored can be of use to improve the quality of existing software and add new perspectives on how to address important limitations.

The introduction of such models adds both an ability to the agent to adapt and possibly improve their actuation over the environment they are inserted in, but also add a certain degree of unpredictability that on other areas of healthcare could result in complex issues. Although most of these virtual agents are still rather limited in learning, adaptation and autonomy, displaying solely reactance to predicted or programmed events, current research methodologies for embedding further intelligence as the proposed open for learning virtual entities. As virtual entities intervene in decision making processes with moral weight, a justified doubt and concern regarding the impact of actions performed by these entities arises. From the numerous scenarios where they can interact with their surrounding environment, some carry moral consequences and describe ethically intricate actions from a human point of view.

Although the main objective was to add a new tool to agent based systems directed towards interoperability, the resulting knowledge isn't only important to these agent based systems. The resulting models reveal tendencies of interoperability procedures within the distinct services and the importance of the distribution of work, which are the cornerstone to understand the usual behaviour of personnel and the flow of information within the information system. Knowledge regarding these subjects is vital to evaluate and improve existing procedures, as well a to detect bottlenecks that undermine the response of the HIS.

Due to this study, the response time of the HL7 services based on multi-agent paradigms and their results were proven to be considerably low. Although no direct comparison can be established, the introduction of intelligent behaviours associated to a good response time demonstrate the potential of this technology towards system interoperability. Furthermore, the recent evolution of HL7 from a syntactic to a semantic paradigm adapts with the perspective of agent-communication paradigm and are an interesting area of current research.

Further techniques must be studied and embedded in current production systems in order to validate their usefulness and the potential to result in agents with the capability to adapt to changes in their environment, perceiving data in a different perspective from human agents. In fact, with the implemented model agents can adapt to changes in the existing environment with far more confidence that system administrators can predict the behaviour of the HIS.

4.1.8 Acknowledgement

This research was performed with the support of the Portuguese Foundation for Science and Technology, with the grant SFRH/BD/65023/2009.

References

Aier S, Schönherr M (2006) Evaluating integration architectures –a scenario-based evaluation of integration technologies. Trends in Enterprise Application Architecture pp 2–14

Berg M (2004) Health Information Management: Integrating Information Technology in Health Care Work. Routledge

Duarte J, Miranda MFM, Abelha A, Santos M, Machado J, Neves J, Alberto C, Salazar M, Quintas C, Ferreira AMSF, Neves J (2009) Agent-based group decision support in medicine. In: Arabnia HR, de la Fuente D, Olivas JA (eds) IC-AI, CSREA Press, pp 115–121

Erl T (2005) Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR

Isern D, Sanchez D, Moreno A (2010) Agents applied in health care: A review. International Journal of Medical Informatics 79(3):145 – 166

Juric MB, Loganathan R, Sarang P, Jennings F (2007) SOA Approach to Integration - XML, Web services, ESB, and BPEL in real-world SOA projects. PACKT Publishing

Kirsh W (ed) (2008) Encyclopedia of Public Health, vol 1. Springer Science

Lopez DM, Blobel BG (2009) A development framework for semantically interoperable health information systems. International Journal of Medical Informatics 78(2):83 – 103, DOI DOI: 10.1016/j.ijmedinf.2008.05.009, URL http://www.sciencedirect.com/science/article/B6T7S-4T08051-1/2/c0d0e805481c88849c83652fa504097b

Machado J, Alves V, Abelha A, Neves J (2007) Ambient intelligence via multiagent systems in the medical arena. ENGINEERING INTELLIGENT SYSTEMS FOR ELECTRICAL ENGINEERING AND COMMUNICATIONS 15(3):151–157

Machado J, Abelha A, Novais P, Neves J (2008) Quality of service in healthcare units. European Simulation and Modelling Conference 2008 pp 291–298, bertelle, C Ayesh, A European Simulation and Modelling Conference OCT 27-29, 2008 European Technol Inst, Havre, FRANCE

Miranda MFM, Abelha A, Santos M, Machado J, Neves J (2008) A group decision support system for staging of cancer. In: Weerasinghe D (ed) eHealth, Springer, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 1, pp 114–121

Ohe K, Kaihara S (1996) Implementation of hl7 to client-server hospital information system (his) in the university of tokyo hospital. Journal of Medical Systems 20(4):197–205, URL http://dx.doi.org/10.1007/BF02263391

PHII (2004) Guiding principles for effective health information systems. Public Health Informatics Institute, Decatur Institute, GA

Weiss (ed) (1999) Multiagent Systems – A Modern Approach to Distributed Modern Approach to Artificial Intelligence. MIT Press, Cambridge, Massachusetts - London, England

4.2 Agent Based Interoperability in Hospital Information Systems

This scientific paper further details the architecture and technical concepts underlying the multi-agent platform uses in order to address many of the limitations described in the specification of the archetype. It also explain the concept of distributed consolidation of information, a core concept of the proposed archetype.

The work contained in this paper is related with the implementation of the system in the *Centro Hospitalar do Porto*.

5th IEEE International Conference on Biomedical Engineering and Informatics

Miguel Miranda, Gabriel Pontes, António Abelha, José Neves and José Machado

Abstract

The healthcare area configures an environment of both complexity and cooperation. Numerous and distinct information systems must exchange information in a expedite and consolidated manner. Where healthcare interoperability is concerned numerous techniques, methodologies, architectures and standards exist, having also some which are more common. However subjects such as service distribution, fault tolerance, standards, communication flavoring and tightly-bound systems still are a major issue of concern. This paper aims to propose and explain a multi-agent based architecture which uses the HL7 standard as a means towards the implementation of interoperability in healthcare environment. It follows the concept of distributed consolidation of information, aiming heterogeneous systems to communicate towards their mutual benefit however through middleware agents which validate and consolidate information.

4.2.1 Health information system and interoperability

Due to its specificities, the healthcare information system configures an environment composed of intricate information technology systems, in which distinct solutions must share data and information consistently and as a whole. The exchange of data and information is of the essence for the optimisation of existing resources and the improvement of the decision making process through consolidation, verification and dissemination of information. Henceforth, within the healthcare environment the integration of all otherwise secluded applications is of the essence for the development of a scalable and functional Health Information System (HIS). The HIS is foremost the consorted and integrated effort of the different heterogeneous solutions within the healthcare institution to collect, process, report and use information and knowledge related to its unique environment to influence the existing management policies, health programs, training, research and medical practice within this institution (Kirsh, 2008). The core concept of a HIS as an abstract global information system for the processing of data, information and knowledge within the healthcare institution, indicates the significance of interoperability between systems in healthcare.

Numerous architectural solutions have been developed towards interoperability in healthcare, depending of the objectives, context and methodological approaches. At this architectural level, one can enumerate distinct and relevant abstract interoperability approaches, such as end-to-end, hub-and-spoke, distributed multi-agent or service oriented. Properties such as modularity, availability, scalability or delay timespan are associated to the interaction of different systems comprised in the devised architecture.

Considering the definition of an HIS, its essence is the architectural model composed of a group of integrated and interoperable solutions within the healthcare institution. In contrast with the usage of a centralized solution, which is unthinkable considering the specificities of each areas of a healthcare unit, it aims to maintain all distinct services and solutions. It is henceforth essential to imbue the HIS architecture with the capacity to allow communication among different and otherwise secluded systems, avoiding their centralisation and dissemination of End-to-End connections, which restrict the growth of all the infrastructure associated to the HIS. The non-modularity of services adds complexity to alterations and improvements, increasing the global costs of the information systems.(Aier and Schönherr, 2006) Therefore, it is understandable the present concern demonstrated by distinct international institutions, responsible for financing and regulating the purchase and development projects for new HIS, with matters of flexibility, interoperation and integration of heterogeneous systems. (Berg, 2004) (PHII, 2004)

Congruently with these concerns, present tendencies regarding research and industry in interoperability applied to healthcare information systems, indicate the potential of agent oriented architecture (Isern et al, 2010)(Machado et al, 2007). Asides from modularity, scalability and adaptability these systems have also the potential to imbue new features associated to intelligent agents which may address the existing problems and solve important limitations otherwise difficult to tackle (Machado et al, 2009). Although healthcare standards like HL7 are completely distinct from agent communication standards, HL7 services can be also implemented under the agent paradigm (Miranda et al, 2010). These agent based HL7 services can communicate with services that follow distinct paradigms and communicate with other agents using either HL7 or agent communication standards. Although the HL7 standard can be implemented using other architectures, agent based solutions enjoy of a vast interoperability capability, being capable to be embedded with the most particular behaviours. These behaviours can become increasingly effective if they use machine learning and other artificial intelligence techniques in order to adapt to the existing environment and being able to prevent errors and correct the flow of information and extraction of knowledge within the institution.

4.2.2 Health Level Seven Protocol

Health Level Seven HL7 gave its first steps as a syntactic healthcare oriented communication protocol at the application layer, the seventh layer of the OSI communication model. The initial versions of the protocol defined the message structure to be exchanged by loosely connected healthcare applications by classifying the different types of messages involved in this environment which were composed at its core by the aggregation of standardized segments.

Henceforth the aim of HL7 is centered on the syntax of what is exchanged, rather than the technology or mean by which this communication occurs nor the underlying architecture. However considering the objective of the communication and the structuring and design of this standard, defining which artefacts of data should be transferred by a certain message and the events which should be be subsequent, the application of client-server architecture was potentiated. In fact, the most common implementation of this architecture using HL7 is based on distinct socket communication clients and servers, in which the client sends an HL7 structured message to the server, that upon processing sends an acknowledgement HL7 standardized message. As mentioned before the HL7 standard is not bound to any technology or either to this architecture, but it is the most widely used in healthcare interoperability.

The initial versions of HL7 were uniquely syntactic, and according to the general models of interoperation are one of the lowest levels of this process. The current version 3 is opening the HL7 scope towards semantic interoperability, including the appropriate use of exchanged information in the sense of the communicating applications behaviour. This model presented in version 3 contains relations and metadata in a abstract level that may enable far higher levels of integration, namely by semantic interoperability and validation of exchanged information, using the relational mapping of each artefact. The Message Development Framework (MDF) is currently moving towards the HL7 Development Framework (HDF), therefore shifting the HL7 paradigm from message to architecture. Newer HL7 developments such as the EHR-S Functional Model and the SOA Project Group activities have been pushing this move (Lopez and Blobel, 2009).

The metadata and archetypes defined in HL7 allow it to organise both production and clinical data in clearly defined and connected segments and fields, which can be validated among artefacts. However, the implementation of version 3 is still rather limited as few service providers and institutions migrated already to this version.

Although version 3 presents several improvements from the previous version, the latter is still the most commonly used in healthcare information systems and equipment. The messages in this version are defined and identified by its control segment. In the control section of the HL7 standard several rules that are applied to all messages are defined:

- Message Segment
- Message Type
- Trigger Events

The core principle underneath he usage of this approach is the principle that behind any practical event there is the requirement for data to flow among heterogeneous systems that comprise the HIS. Henceforth, most events on the healthcare environment act as triggers for the initiation of information dissemination. While an event can emerge at one system and be handled by this system alone, being the flow of information to other ones aimed mostly at maintenance of consistency; an event can be initiate at one system but need to be handled by another, in which case the information transaction is named an unsolicited update. The scope of the standards aim is to solely specify messages between systems and the events which trigger them. No considerations regarding underlying systems architecture and implementation are concerned by HL7.

A trigger event may come from one of the following sources:

User Request Based (in this document also referred to as Environmental) - For example, the trigger event that prompts a system to send all accumulated data to a tracking system every 12 hours is considered Environmental. Similarly a user pressing a button in a user-interface would be considered environmental

State Transition - resulting from a state transition as depicted in the State Transition Model for a particular message interaction. The trigger for cancelling a document, for example, may be considered a State Transition Based trigger event

Interaction Based - based on the receipt of another interaction. For example, the response to a query (which is an interaction) is an Interaction Based trigger event.

From this perspective the flow of information between al information systems and elements in the healthcare institution, or by other words the entire HIS is governed by these events. Its the paper of this standard to regulate ad defined all these events as well as their implication and required information for the underlying procedures.

The fact that most of the communications are currently being performed with a syntactic and flavored norm such as HL7 version 2 results in loss of modularity and inherent stronger coupling than desirable between systems. Moreover the complexity of flavoring and specificities of each interoperation among systems restricts a standard and extended evaluation of the information within the message before disseminating it among all the systems which compose the HIS.

Although other standards and technologies such as HL7 version 3 or openEHR allow further semantic reasoning and validation, its implementation in real environment is far from a solution considering the difficulties of dealing both of legacy systems or flavored approaches. Moreover the efficiency of the piped HL7 under version 2, and its optimisation over the years turns the migration to these more complex standards and technologies (e.g. XML) far more intricate as it can result in a overhead in equipment and systems communication. A rather more technological but also significant source of problems is the existence of unpredictable communication failures in the network associated mainly with the hours of higher production rate.

4.2.3 Multi-agent systems for interoperability

The multi-agent system paradigm has been an interesting technology in the area of interoperability in healthcare. Being multi-agent architectures a field of research of distributed artificial intelligence, this technology is intrinsically connected to the basilar concepts that define a distributed architecture, while being distinct in the intrinsic definition of an agent versus the properties of the general middle-wares of many others distributed architectures. Being distributed by nature its characteristics introduce MAS as a rich and fiercely adaptable technology with great interest mainly due to the research interests in this arena.

In an environment where the demand for middle-wares both for production and legacy systems are constant, the agent paradigm demonstrates an intuitive advantage in organizational development in terms of creation of such services.

In this research area concepts and technologies such as terminologies, ontologies, mobility, failure recovery and intelligent behaviours are embedded or explored in many existing frameworks. Henceforth they of interest for healthcare interoperability and a tool towards intelligent interoperability systems.

4.2.4 Hl7 services in multi-agent system

As mentioned before the HL7 standard does not limit the its usage to any technology or architecture, however being the objective of its usage to regulate communications in healthcare oriented systems, there are obviously technologies and architectures that became the most used. Henceforth, the technologies and architectures that grew more common are the ones that are present by default from information systems to specific equipment for the execution of diverse complementary diagnosis methods.

However exchange of communication is not solely limited to occur between information systems, as communication with equipments is ever-growing more important. This fact is very important to consider since it implicates that not only information systems are concerned with standards and technologies when dealing with communication, the equipment must as well deal with such characteristics. This equipment usually either communicate through the usage of standards in a loosely-coupled manner, i.e. directly with an information system (Radiological Information System, Cardiological Information System, ...) or with a proprietary system which can in its term be compatible or not with other information systems. This sort of equipment usually follows a client/server architecture, in which the equipment is is most cases solely a client.

From what is understandable from these last paragraphs there is a considerable difficulty in creating a system that uniformly understands and communicates fully with all services within a

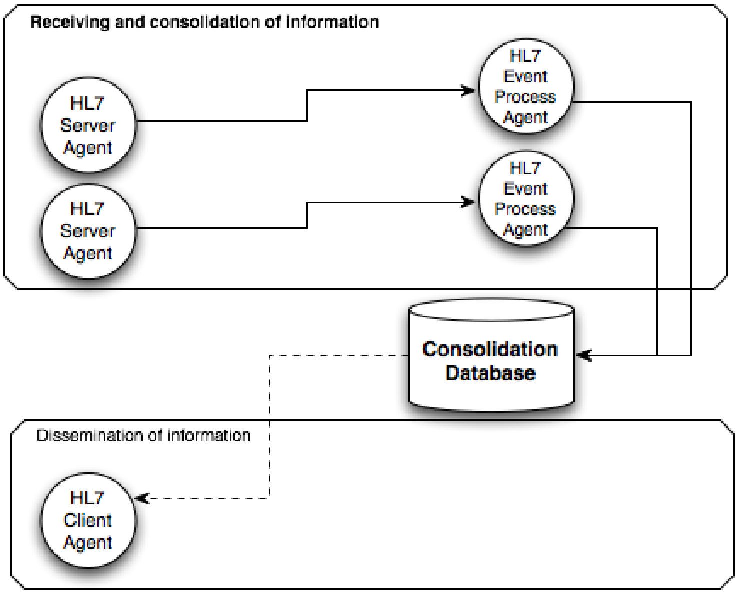


Fig. 4.3: Information flow inherent to the developed HL7 service using a multi-agent system

HIS. As explained before the even with the overall adoption of standards, more specifically HL7, different flavorings usually require a distinct handling of messages and its events.The unique characteristic of the agent parading allows to create specific behaviours or agents which are adapted to any situation while keeping all systems loosely coupled.

The currently detailed multi-agent system was developed under WADE/JADE and is currently in production in mid-sized regional hospital, being responsible for the consolidation and distribution of information in this environment. It is responsible for more than processing and disseminating HL7 messages, it perform several back-office functions indispensable for the functioning of the HIS. However, for the scope of this paper only the HL7 related functionalities will be elaborated.

The concept aimed to represent and implement through this multi-agent system is the idea of distributed consolidation of information. This agent system consolidates in its own data model the information considered relevant for all information systems. When a HL7 event is received through any server agent, it is pre-processed and forwarded to another agent ready to handle this message and its events. The consolidation of this information generates events that are disseminated as HL7 events and messages throughout the systems that are registered as servers in the parametrization of the multi-agents system.

An HL7 server agent may receive messages from several clients, depended or wether or not the client keeps the connection open even while not sending messages or the number of ports open for the TCP pipe parser. As each agent by default in JADE is a unique thread, a sound characteristic that follows the perspective of an agent identity, a single socket pipe parser is preferred per agent, however the pre-processing of each message is handled as

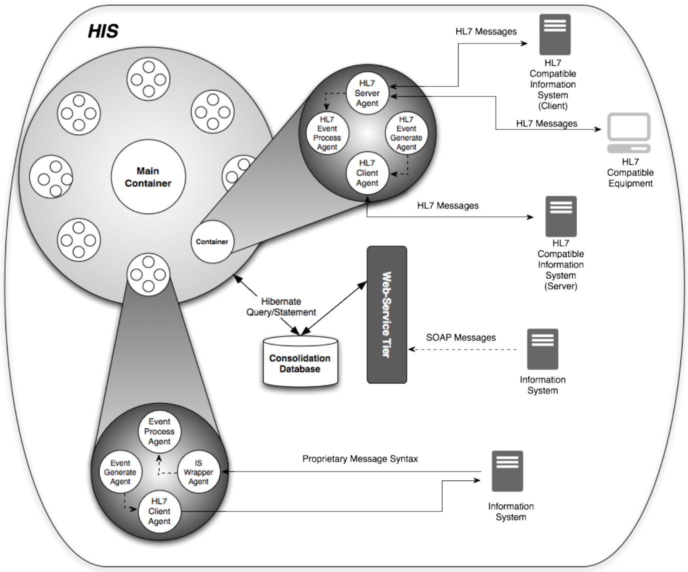


Fig. 4.4: Overall architecture of the HL7 communication using a multi-agent system

a new thread by feeding the message to a behaviour which is encapsulated in a Threaded-BehaviourFactory object. This methodology is preferred not to lock the agents life-cycle, to process the message immediately and to grant that the socket is freed as soon as possible.

As demonstrated in Figure 4.4 there are several containers with agents to receive and process messages, however the organisation per container is not significant. Analyzing the server agent in greater detail, its behaviours are orientated solely with the forwarding of a specific message and event to an agent prepared to provide these services. According to parametrisation of the MAS, the messages are forwarded to agents responsible for dealing with the service which sent the message and the specific event. Such agents are easily found by the use of a common ontology that defined not only agent communication but also details pertaining the services registered at the directory facilitator, so that the server agent can easily find the agent the message must be forwarded to. This dynamic information flow allows to have a set of genera processing agents as well as to add and remove specific agents when needed for a specific interoperability service. Moreover throughout this methodology only the server agents need to be fixed to a specific container and machine, the agents that process the messages and have most of the workload, can have mobility and move from machine to machine, container to container as necessary and also be created more than one agent for the same description service, distributing load and enabling failure prevention.

The consolidation of information is achieved by the conversion of HL7 events to an ontology that also represents the consolidation model present in the relational database. This allows the ontology objects to be handled directly in synchrony and congruently with the database.

For this purpose the basic JADE framework as extended with a hibernate implementation significantly different from the standard made available.

From the processing and consolidation of received events the need usually occurs for client agents to communicate with other information systems in order to disseminate it. These tasks for the client can either emerge from requests by agents that process and consolidate the information resulting from an event or form the consolidation process itself. The relational database when altered possesses a set of triggering for certain events according to parametrization, which may create tasks for the client agents.

This mentioned architecture in Figure 4.4 is oriented towards the concepts of dissemination and consolidation of HIS information in order to synergistically improve the quality of all information system involved. The concept aimed to represent and implement through this multi-agent system is the idea of distributed consolidation of information. This agent system consolidates in its own data model the information considered relevant for all information systems. The consolidation of this information generates events that are disseminated throughout the systems that are registered as servers in the parametrisation of the multi-agents system. From the processing and consolidation of received events the need usually occurs for client agents to communicate with other information systems in order to disseminate it. These tasks for the client can either emerge from requests by agents that process and consolidate the information resulting from an event or form the consolidation process itself. The relational database when altered possesses a set of triggering for certain events according to parametrisation, which may create tasks for the client agents to communicate with others.

Considering this architecture there is obviously a shift from the usual end-to-end architecture, in which services directly intertwine with each other creating a complex mesh of connections. This architecture aims to turn the services available at the HIS loosely-coupled to an extent that adding or removing a service is a matter of changing parametrization within the multi-agent system or add a specific agent type that handles communication translation to a standard one. If a information system is removed or fails to respond the meaningful information remains stored at the consolidation database, while events that will consolidate the system with the rest of the his are being stored and scheduled in this same consolidation database.

4.2.5 Conclusions

The usage of multi-agent systems in interoperability problems constitutes a significant research opportunity to improve the communication among heterogeneous systems. Several of the research interests of agent technology such as ontologies, mobility and fault tolerance among many others can be of great use and interest to be applied in this area.

Although this module represents solely a part of greater project aimed towards a HIS with enhanced forms of interoperability, it is of great significance and interest as HL7 is the most common standard for healthcare communication among heterogeneous systems. This module is currently under validation being introduced gradually into the production MAS.

The most important characteristic of this architecture and model is that instead of a mesh of end-to-end system communication or a major centralisation of processing, this paradigm is by nature distributed but allows a consolidation of processual and clinical validation of information. This consolidation is of the essence for the the establishment of a complete electronic health record in an environment in which heterogeneous information systems exist.

Acknowledgement

This work is funded by National Funds through the FCT - Portuguese Foundation for Science and Technology within project PEst-OE/EEI/UI0752/2011. Work of Miguel Miranda is supported by FCT with the grant SFRH/BD/65023/2009.

References

Aier S, Schönherr M (2006) Evaluating integration architectures –a scenario-based evaluation of integration technologies. Trends in Enterprise Application Architecture pp 2–14

Berg M (2004) Health Information Management: Integrating Information Technology in Health Care Work. Routledge

Isern D, Sanchez D, Moreno A (2010) Agents applied in health care: A review. International Journal of Medical Informatics 79(3):145 – 166

Kirsh W (ed) (2008) Encyclopedia of Public Health, vol 1. Springer Science

Lopez DM, Blobel BG (2009) A development framework for semantically interoperable health information systems. International Journal of Medical Informatics 78(2):83 – 103, DOI DOI: 10.1016/j.ijmedinf.2008.05.009, URL http://www.sciencedirect.com/science/article/B6T7S-4T08051-1/2/c0d0e805481c88849c83652fa504097b

Machado J, Alves V, Abelha A, Neves J (2007) Ambient intelligence via multiagent systems in the medical arena. ENGINEERING INTELLIGENT SYSTEMS FOR ELECTRICAL ENGINEERING AND COMMUNICATIONS 15(3):151–157

Machado J, Miranda M, Abelha A, Neves J, Neves J (2009) Modeling medical ethics through intelligent agents. In: Godart C, Gronau N, Sharma S, Canals G (eds) Software Services for e-Business and e-Society, IFIP Advances in Information and Communication Technology, vol 305, Springer Berlin Heidelberg, pp 112–122

Miranda M, Pontes G, Gonçalves P, Peixoto H, Santos M, Abelha A, Machado J (2010) Modelling intelligent behaviours in multi-agent based hl7 services. In: Lee RY (ed) Computer and Information Science, Studies in Computational Intelligence, vol 317, Springer Berlin, pp 95–106

PHII (2004) Guiding principles for effective health information systems. Public Health Informatics Institute, Decatur Institute, GA

4.3 Evolutionary Intelligence in Agent Modeling and Interoperability

This study provides an overview of the agent based model that is prosed in this thesis. It details important aspects of the introduction of ontologies as part of agent communication towards interoperability and the quality of information exchanged among heterogeneous systems.

Some important technical points regarding JADE and WADE are mentioned in this study, namely addressing the mapping between HL7 schemas in XSD format for XML exchange within agent ontology objects.

Advances in Intelligent and Soft Computing, 2011, Volume 92/2011, 253-257

Miguel Miranda, José Machado, António Abelha, José Neves and João Neves

Abstract

A healthcare organization to be tuned with the users expectations, and to act according to its goals, must be accountable for the quality, cost, and overall care of the beneficiaries. In this paper we describe a model of clinical information designed to make health information systems properly interoperable and safely computable, based on an Evolutionary Intelligence approach that generates quantified scenarios from defective knowledge. The model is a response to a number of requirements, ranging from the semantic ones to the evaluation of software performance at runtime; it is among the biggest challenges in engineering nowadays.

4.3.1 Introduction

Considering the course and the initial impetus on agent-based systems and agent-based methodologies for problem solving, it is understood that it is not available, yet, a real bunch of practical applications and implementations outside the realm of research in software analysis and development. However, agent-based systems are arriving at a stage of maturity, i.e., there are tools such as JADE, WADE and others frameworks that started a consensus that may help to add agent-based systems to the main stream of programming methodologies for problem solving.

When approaching a problem, any interpretation cannot be bonded, *a priori*, to a specific technology or methodology for problem solving. New systems must be oriented and adapted to better fit into the environment. Using multi-agent systems or agent-based models for problem solving presents a different way to set different courses in research methodology and practice, i.e., a system of methods used in a particular area of study or activity, that can be imbedded in an organized manner.

On the other hand, a Service Oriented Architecture (SOA) denotes, fundamentally, a collection of services, which communicate with one another. The communication process can involve either simple data going past or across or it could include two or more services negotiating some activity; where a service is to be understood as a function that is well-defined, self-contained, and does not depend on the context or state of other services. Connecting services is therefore paramount; indeed enterprise applications and software systems need to be interoperable in order to achieve seamless business across organizational boundaries. It may seem extremely similar to the MAS approach, however SOA is an abstraction not bound to one specific technology, once it can be based on web-services, agents or in any other framework following these basic rules. The same can be applied to ontologies for cross-organizational communication, which have been already integrated in existing agent development frameworks. Indeed, new business models also call for innovative approaches to customers, involving collaboration across different organizations and domains and therefore need cross-organizational communication.

4.3.2 A Multi-Agent Environment for Distributed Interoperability in Healthcare

In order to enable increased levels of interoperability, a whole environment centered in agent-based systems was developed to take advantage of the agent based methodology for problem solving. As it is visible in Figure 4.5, the applicational environment can be divided according to the following tiers of abstraction: Multi-Agent Distributed Interoperability Platform (MADIP); Web-based Interoperability Platform (WIP); Hibernate and the Relational Database Management System.

The external service providers depicted in Figure MAEDIH include different information systems such as the Radiological Information System, Laboratory Information System, Hospital Emergency Rooms Information Systems and other back-end and front-end to external applications. This environment went in production at the Centro Hospitalar do Porto, EPE, a major healthcare facility in the north of Portugal, as an element of the agency AIDA (Figure 4.6) (i.e., an Agency for the Integration, Diffusion and Archive of Information), that is in charge of the interoperability among all the services present in a healthcare environment, here depicted as a Healthcare Information System (HIS). A HIS can be defined as an abstract information system for data processing within an healthcare institution. It is therefore the consorted and integrated effort of the different, tentatives, heterogeneous information systems inside the healthcare institution, which collects, processes, reports and use information and knowledge within this unique environment, i.e., that influences the existing management policies, health programs, training, research and medical practice within the institution (Kirsh, 2008) (Miranda et al, 2010).

4.3.2.1 Core Architectural Dependencies

Regarding the MADIP architecture, that is a multi-agent system developed under the Workflows and Agents Development Environment (WADE) and the Java Agent DEvelopment Framework (JADE). This choice was taken considering that JADE is an Open-Source JAVA based solution which offers some advantages, namely the production environment tests, a continuous improvement in any new release, and the demonstrated stability that assures the support and capability to handle all the events that may occur in a production environment. Furthermore, the fact that the framework is fully compliant with the specification of the Foundation for Intelligent Physical Agents (FIPA) using the advised architecture and making available several agent communication codecs such as FIPA Semantic Language (SL) and XML, enables one to extend the existing framework to a particular setting, without "reinventing the wheel" (Bellifemine et al, 2010). Complementary to the Agent Communication Language (ACL), the existing ontology server and ontology embedded communication trough plain Java objects, constitutes an important asset that is made available by JADE.

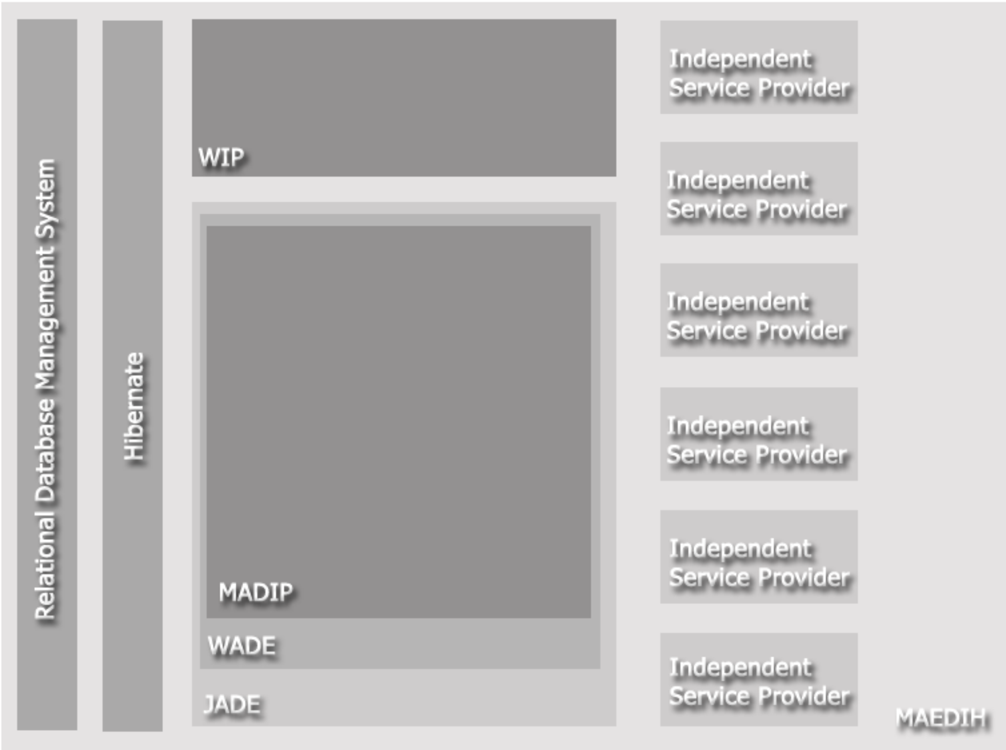


Fig. 4.5: Core tiers of the integration environment.

4.3.2.2 Interoperability Interfaces in Agent Modeling

Diverse methodologies for problem solving may be employed to achieve integration and interoperability. They are usually comparable according to their specificities and the necessities on information availability, quality, quantities and processing time.

In order to interact with different service providers that do not use ACL with FIPA compliance, the most common interoperability methodologies for problem solving are put into effect through configurable agents, using the methods: Health Level Seven (HL7) *via* TCP socket; HL7 *via* SOAP and Web-Services; AIDAs XML *via* SOAP and Web-Services and Legacy System via TCP socket. Using plain Java objects as mappings for the XML and HL7 structures being provided, it is possible to associate them to diverse medical ontologies, here given in terms of Knowledge Bases (KBs), i.e., formal systems that capture the meaning of the adopted vocabulary via logical formulas. A KB is considerably richer than a conceptual schema since the underlying

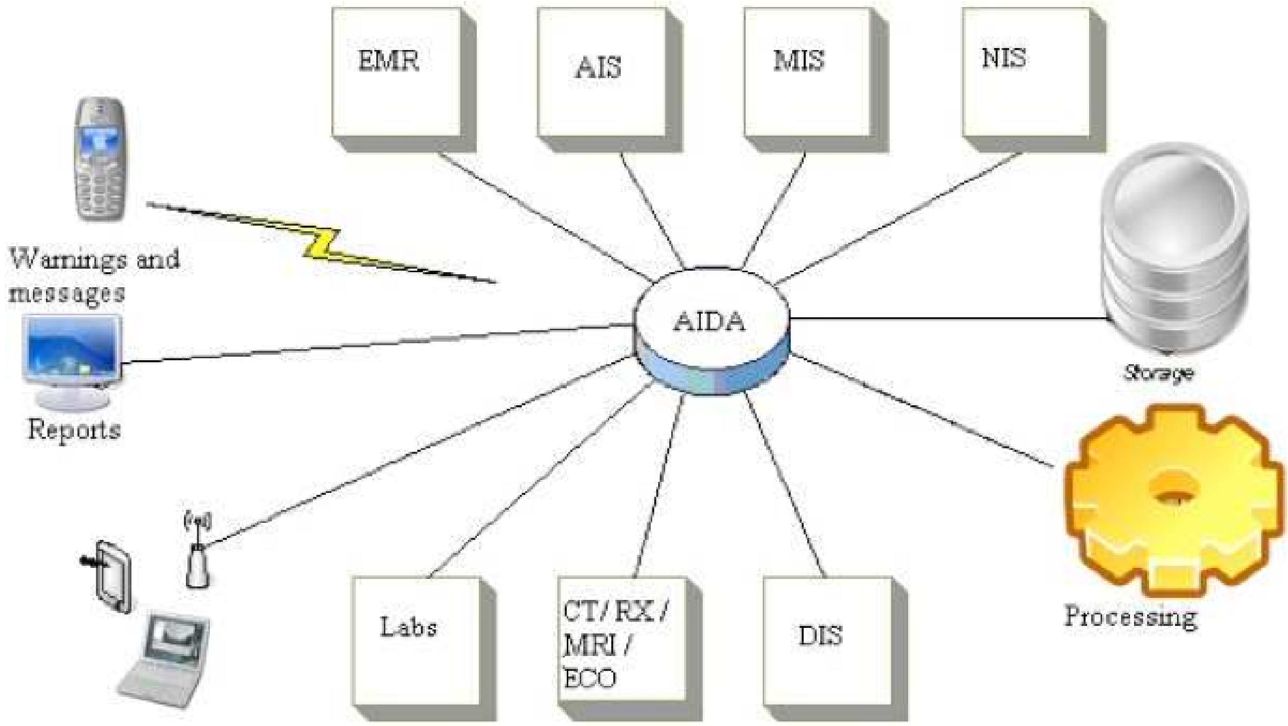


Fig. 4.6: AIDA platform

languages are more expressive. The purpose is not simply retrieval but reasoning. However, the main task is still data consistency. Formally, a knowledge base is a Logic Theory. Communication within and among agent-based platforms, considering the necessity of interaction and exchange of information, is without a doubt a case of interoperability.

4.3.2.3 Interoperability Enhancements through the use of Ontologies

When interoperability is accounted in terms of the nature of being, i.e., in terms of an or more ontologies, it has the potential to improve the flow of information and knowledge exchange inter peers. Tis is a good or obvious cause to use the WADE/JADE based ontologies to manage content expressions as Java objects across domains and applications (Ghosh et al, 2010). Although they are handled as Java objects, they are encoded and decoded into messages in a standard FIPA format. The ontology engine in JADE is made on a name, a running ontology that is susceptible of a particular figure or combination and a set of element schemas which may contain 3 (three) basic interfaces, namely Agent Action, Predicate Extensions and Concepts(Caire and Cabanillas, 2010), i.e., these element schemas are objects describing the structure, semantics and relationships among the Agent Action, Predicate Extensions and Concepts that hold within a given ontology, which may share vocabulary among the different agents using it. Rather than the Ontology, BasicOntology and ACLOntology, the BeanOntology can be used to map JavaBeans to an ontology using a "convention over configuration" perspective (Cancedda and Caire, 2008). By this mean JavaBeans annotations are used to extend the objects that constitute the ontology. The Agent Action denotes a content expression of a direct request among agents to perform a given task. This action request will contain the information regarding what action and to what objects it is to be applied. For this reason all requests received from any service provider were transformed through object mappings into an Agent Action that was communicated to the agents involved in the process. Whenever an agent asks for the truth value of a given sentence, the sentence can be defined as a class that implements a Predicate Extension(Caire and Cabanillas, 2010).

In the proposed ontology, the definition for interoperability, Agent Actions and Predicates very often holding Concepts. A class that implements a Concept interface is composed of several objects implicitly conjuncted. Existing mapping using Hibernate are also compatible to be mapped within these concept classes making it easy to exchange information between agents while the content is completely normalized and congruent with the storage schema. The use of ontologies over serialized objects allows one not only have interoperability with other agent-based systems, but, to a some extent, to have interoperability with further non agent-based systems.

4.3.3 Conclusion

This environment was tested in the first place and put into production in an ambulatory surgery centre as a part of a huge healthcare facility. The need to interoperate was felt at almost all levels of decision as defined by the business logic, which is not part of this document. We argue that the starting point of a successful model must be an ontological analysis of the process of clinical care delivery, seen as a scientific problem-solving process. The combination of Artificial Intelligence techniques such as the symbolic and evolutionary systems allows us to combine the advantages of each of these approaches, in particular for solving complex problems. These techniques are based on the collective adaptation and learning ability of individuals.

References

Bellifemine F, Caire G, Trucco T, Rimassa G (2010) Jade programmer's guide. Tech. rep., Telecom Italia S.p.A.

Caire G, Cabanillas D (2010) Application-defined content languages and ontologies. Tech. rep., Telecom Italia S.p.A.

Cancedda P, Caire G (2008) Ade tutorial creating ontologies by means of creating ontologies by means of the bean-ontology class. Tech. rep., Telecom Italia S.p.A.

Ghosh K, Natarajan S, Srinivasan R (2010) Decision fusion in distributed multi agent process supervisory system. Networking, Sensing and Control (ICNSC), 2010 International Conference on pp 183 –188, URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5461510

Kirsh W (ed) (2008) Encyclopedia of Public Health, vol 1. Springer Science

Miranda M, Pontes G, Gonçalves P, Peixoto H, Santos M, Abelha A, Machado J (2010) Modelling Intelligent Behaviours in Multi-agent Based HL7 Services, vol 317, Studies in Computational Intelligence, Springer Berlin / Heidelberg, pp 95–106

4.4 Step Forward Medical Ethics Modeling

When developing intelligent agent systems to be applied in areas of moral complexity such as in healhare, where intrisically invaluable and imensural compensation items such as life and health have to be taken into account, the theme of how to deal with such conflicting should come into consideration. For this purpose and from a theoretical point of view, this study discusses this complex theme and proposes a model based on continuous as a paradigm for the expression and resioning regarding moral cenarios.

Although this model was demonstrated in this study and had a strong acceptance by the international scientific comunity, due to technological incompatibilities with the core systems developed and implemented this has not yet been implemented in any of the protpotypes.

The results of this research are very important for future work in this area and proff themselves of great value not only for the the prospect of agents with greater intelligence and resonsability in healthcare interoperability, but also to comprehend how moral reasoning can be modeled in computational environments.

IFIP Advances in Information and Communication Technology, Volume: 335, Springer, 2010

Miguel Miranda, José Machado, Gabriel Pontes, António Abelha and José Neves

Abstract

Modeling of ethical reasoning has been a matter of discussion and research among distinct fields, however no definite model has demonstrated undeniable global superiority over the others. However, the context of application of moral reasoning can require one methodology over the other. In areas such as medicine where quality of life and the life itself of the patient are at stake, the ability to make the reasoning of the models understandable to staff and to change their reasoning according to paradigm and context change is of the essence. In this paper we present some of the modeling lines of ethical reasoning applied to medicine, and defend that logical programming presents characteristics, which are essential for the purpose of trustworthy morally aware decision support systems. It is also demonstrated a model for moral decision on two cases at Intensive Care Units, a service where the moral complexity of regular decisions is an incentive for the study and development of moral decision support methodologies.

4.4.1 Introduction

Overviewing the evolution of technology and information systems, a trend of growing pro-activeness and limited intelligence is pushing the role of virtual entities, step-by-step, higher. Many activities are nowadays performed by automated entities, while supervised by humans. Although most of these virtual entities are still rather limited in learning, adaptation and autonomy, displaying solely reactance to predicted or programmed events, several threads of Artificial Intelligence research methodologies for imbedding further intelligence.

The notion of virtual entity is here used to differentiate entities with higher levels of autonomy, learning, prediction and decision from a mainly reactive and controlled machine. Moreover, considering developments in the area of informatics and Artificial Intelligence (AI) in particular, it must be considered that many of these entities can exist within a single physical machine or even that a single one can be distributed within limitless machines. Therefore the notion of a virtual entity in this case is similar to the concept of an agent in the area of Multi-Agent Systems.

As virtual entities become more complex and hold critical functions, a justified doubt and concern regarding the impact of actions performed by these entities arises. From the numerous scenarios where they can interact with their surrounding environment, some carry moral consequences and describe ethically intricate actions from a human point of view. From the need to prevent immoral decisions and ensure confidence regarding these virtual entities, further understanding of the capacity of moral agency, moral modeling and the complexity human moral ethics.

Modeling machine ethics can result in further understanding of human ethics itself, either by defining rules and exceptions, or by knowledge extraction, case classification and patterns search over existing cases and outcomes using different algorithms. One can in fact consider that from the numerous methodologies that exist for the study of moral capacity, for each of them different subsequent potential outcomes can be found. While modeling ethics based on defined moral principles can help defining ethical principles and validate the resulting decision process, using learning algorithms and knowledge extraction over existing moral cases and outcomes can deepen the understanding of the underlying moral rules and patterns that may go unnoticed, but define moral decisions. In other words, theses processes aiming to analyze the essence of morality can be used not only to study their simulation/emulation, but also to deepen and evaluate the moral standards and dilemmas in ethically complex systems. The results from these systems are not limited outcome decisions before an ethical complex problem. Using a perspective of decision support or decision optimization, from a knowledge-base (either by previous studied cases or expert input), bearing in mind a specific scenario, similar cases can be aggregated for human user consideration, rules/principles involved in the decision can be induced with a certain degree of certainty, or conditions can be abduced.

There exists no definite solution for modeling ethical virtual entities, and presently several approaches are being presented and some compared against one another. Studying the present study and investigation in the area, different methodologies for modeling moral capabilities using artificial intelligence techniques can be segmented according to their main characteristics (Tonkens, 2009). One of the most definite and important disparity in methodologies is the usage of explicit reasoning versus black-box reasoning. In explicit reasoning, the processes underneath a moral decision are clearly defined as principles, rules, exceptions, or other structure defined for one particular modeling. When analyzing AI techniques derivatives of symbolic, sub-symbolic or statistical approaches, there exist some that are able to represent their "line of though", allowing a transparent view of the moral decision process (Nugent and Cunningham, 2005).

One of these techniques is logic programming, in which horn clauses contain the formalisms that mold the reasoning within an existing logical predicate. Current research indicates that non-monotic logic, due to its ability to implement defeasible inference, enabling moral principles to add and still diminish the set of conclusions determined by the knowledge base, is an interesting and promising technique to model moral reasoning (Horty, 1994) (Powers, 2006) (Machado et al, 2009). By this mean, principles of benevolence and non-malificience can exist in accordance with other principles that are against their value or state an exception for superseding context principles. Regardless of the use of deductive, inductive or abductive logic, the rules used or attained are explicitly defined. However, the usage of each of these techniques of logic programming varies on the objective and context of application.

On the other hand, while using black-box reasoning, the reasoning behind the moral decision itself cannot be perceived in a clear manner. In other words, within the process of a black-box technique, facing a set of inputs, only a set of outputs can be obtained, not the process or reason behind it. That is the case of neural networks, regardless of the methodologies used to attempt to understand the reasoning behind them, the fact remains that no certainty of the processing underneath the trained neural network exists (Nugent and Cunningham, 2005). Although interesting results can be achieved using neural networks trained on existing moral cases and consequently implementing case based reasoning, the understanding of the moral principles within these black boxes is unknown (Guarini, 2006). Different techniques can be used to reverse-engineer neural network's inner structure and imbedded rules, however, the result is not exactly the rules used but rather an induced or a probabilistic set of them (Floares, 2007). In the end of this reverse-engineer process, it is attained an induced set of rules of a systems that already uses induction or probabilistic methods to train its processing, revealing a certainty of doubt over the extracted rules.

Another divergence in ethical modeling is the learning process of rules or reasoning methodologies in ethical dilemmas. When considering a specific area such as medicine, most of the existing knowledge essential to model moral reasoning is contained in deontological principles or case studies (Jonsen et al, 1997). In either of these cases the core of this knowledge is based on individuals or panels of experts. In light of these sources, the moral decision model can be developed from existing principles, from learnt principles or from hybridization of both sources.

While one can consider existing deontological principles as existing principles, learnt principles are those extracted from existing cases. These machine-learning behaviors applied to ethics are a rather complex theme as principle learning may result in immoral principles and depending of the methodology used it may not be possible to clearly understand the underlying principles (e.g. black-box machine learning). Inductive logic programming has also expressed in existing research potential to induce principles and their relations from experts reasoning.(Anderson et al, 2006) When modeling moral behavior in virtual entities, researchers must always bare in mind the environment that molds its principles. For research purposes selection of an area and a purpose is of the essence in order to evaluate results and contextualize the used approach. With this in mind, the disparity between ideal and real environments in the medical arena creates a complex set of scenarios, which are pressing and interesting to analyze from an ethical point of view.

Therefore, this article will address moral reasoning in medicine, and apply it in clinical context.

4.4.2 Medical ethics modeling - analysis and applications

Clinical ethics is an arena of public interest, where themes such as end of life, abortion and refusal or futility of treatment, among others, are constantly discussed as specific dilemmas occur or opinions and believes change. Although the deontic principles of a physician remain centered in the best practice towards the patient, legislation and court decisions mold the parameters of how physicians should behave in specific cases, which bare moral consequences. In fact, the context in which a morally complex case presents itself may uphold different results.

One european study analyzing the frequency and types of withholding and withdraw of life-sustaining therapies within the Intensive Care Units (ICU) of European countries, indicated that different countries and cultures deal in diverse ways with ethical dilemmas arising from these therapies (Sprung et al, 2003). One can go a step further, and consider the hypothesis that the physicians training and context can as well affect the moral decision making process. In fact, these decisions of withholding and withdraw of therapy, similarly to many other in clinical ethics, are far from an hypothetical situations, they happen frequently in the healthcare arena and allow no time for extensive legal or ethical consulting by the physician responsible for this decision. The moral demanding of clinical staff is overwhelming and can become even more complex and dubious in contexts of intensive and emergency care. Intensivists are constantly presented with new moral dilemmas, which demand for a quick and asserted answer (Danbury and Waldmann, 2006). Medical staff must therefore, be taught and trained to deal with these situations. Studies analyzing moral dilemmas and ethics modeling methodologies can be of help in this mater, to enhance the existing guidelines and understanding of moral decisions.

In the area of medicine, both practice and research activities have been actively overviewed and ultimately limited by existing legislation and court jurisprudence. This legislative effort is deeply connected to the existing moral principles and ethical concerns (Danbury and Waldmann, 2006). However, the existing legal directives can ease a decision concerning a morally complex situation and ethical confrontation, without fear for civil consequences. Some limitations occur on situations, in which decisions that sound ethically sound are limited by law, nevertheless professional conduct codes generally defined the proper conduct within the limits of the law (Jonsen et al, 1997).

For centuries, the clinical ethics with roots on Hippocrates principles defined as its main deontological fact, the obligation of the physician to give to the patient all treatments medicine knowledge considered the best fit. Nowadays, the decision is centered on the patients will, moreover, with the development of medical technologies, through their breakthrough and short-comes, physicians have also to take in consideration consequences of physical, mental and financial order (Serrão and Nunes, 1998). This change of paradigm and the subjacent increment of ethical and civil load to the decisions of clinical staff, is an environment where synergies of medical ethics and AI, in order to understand how moral processing should be designed and how tutoring and decision support systems can be developed and implemented.

One interpretation of the process of learning and practicing clinical ethics is based on a set of corner-stones rules (i.e. moral principles), completed by the interpretation of existing fact in light of the existing numerous case studies. One can therefore consider that the moral behavior of physicians is a complex intertwined system of both rule-based and case based reasoning. Case studies can represent to some extents either rules or specific conditions which classify exceptions. This notion of exception is one of a logical programming point of view, where a context of known and unknown values of an universe can result in an exception to an existing predicate. Case studies can concur with the existing moral principles, alter their relationship, or define a context in which the existing principles were disregarded. When one analysis an ethical case study in medicine, the surrounding context that materializes the moral action defines an example of a decision with moral consequences, where the boundaries of right or wrong are complex to ascertain. The analysis of such cases is complex, however one should always bear in mind that the existing moral rules and principles of medicine are the barebones of clinical ethics and should not be superseded unless valid exceptions are deemed correct.

From the distinct environments within the medical arena, intensive care medicine embodies an environment where moral decisions are usual and complex. In this specific context, decisions must be taken within short time spans while also regard limited resources and patients in critical conditions (Danbury and Waldmann, 2006). This context enables interest in using moral decision modeling in clinical cases appertaining to the ICU.

4.4.3 Modeling clinical ethics

With respect to the computational paradigm, it was considered Continuous Logic Programming (CLP) with two kinds of negation, classical negation, $\neg$, and default negation, not. Intuitively, following the close world assumption, not p is true whenever there is no reason to believe p, whereas $\neg p$ requires a proof of the negated literal. A continuous logic program (program, for short) is a finite collection of rules and integrity constraints, standing for all their ground instances, and is given in the form:

$p \leftarrow p_1 \wedge \ldots \wedge p_n \wedge$ not $q_1 \wedge \ldots \wedge$ not q_m; and
$?p_1 \wedge \ldots \wedge p_n \wedge$ not $q_1 \wedge \ldots \wedge$ not $q_m, (n, m \geq 0)$

where $?$ is a domain atom denoting falsity, the p_i, q_j, and p are classical ground literals, i.e. either positive atoms or atoms preceded by the classical negation sign $\neg$. Every program is associated with a set of abducibles. Abducibles may be seen as hypotheses that provide possible solutions or explanations of given queries, being given here in the form of exceptions to the extensions of the predicates that make the program. Therefore, being Γ a program in Extended Logic Programming (ELP) and $g(X)$ a question where X contains variables $X_1 \wedge \ldots \wedge X_n (n \geq 0)$, one gets as an answer:

The answer of Γ to $g(X)$ is $true$ iff
$g(X) \rightarrow demo(\Gamma, g(X), true)$.
The answer of Γ to $g(X)$ is $false$ iff
$\neg g(X) \rightarrow demo(\Gamma, g(X), false)$.
The answer of Γ to $g(X)$ is $unknown$ iff
not $\neg g(X) \wedge$ not $g(X) \rightarrow demo(\Gamma, g(X), unknown)$.

where $unknown$ stands for a truth value in the interval $0...1$. Being Γ a Program it is possible to define the Minimal Answer Set of Γ $(MAS(\Gamma))$:

$\Gamma \vdash$ s iff s $\in$ MAS(Γ)

where $\Gamma \vdash$ s denotes that s is a logical consequence or conclusion for Γ.

Being now AS_i and AS_j two different answer sets of Γ, being EASi and EASj, respectively, the extensions of predicates p in ASi and ASj, it is defined that ASi is morally preferable to ASj ($AS_i < AS_j$) where < denotes the morally preferable relation, denoting that for each predicate $p1$ there exists a predicate p2 such that p1 < p2 and $EAS_i \backslash EAS_j$ is not empty (denotes the difference set operator).

In our approach, the morally preferable relation is based on evolution and it is built on a quantification process of the quality-of-information that stems from a continuous logic program. Indeed, let p_i $(i \in 1, \ldots, m)$ denotes the predicates whose extensions make a continuous logic program that models the universe of discourse, in terms of the extensions of predicates and let a_j $(j \in 1, ..., n)$ stands for the attributes for those predicates. Let $x_j \in [min_j, max_j]$ be a value for attribute a_j. To each predicate it is also associated a scoring function $V_i j[min_j, max_j] \rightarrow 0...1$, that gives the score of predicate p_i assigned to a value of attribute a_j in the range of its acceptable values, i.e. its domain (for sake of simplicity, scores are kept in the continuous interval $[0, ..., 1]$. The quality-of-information with respect to a generic predicate it is therefore given by $Q_i = 1\frac{1}{Card}$, where Card denotes the cardinality of the exception set for the predicate p_i, if the exception set is not disjoint.

If the exception set is disjoint, the quality of information is given by $Q_i = \frac{1}{C_1^{Card} + ... + C_{Card}^{Card}}$ where C_{Card}^{Card} is a k-combination subset, with card elements. The relative importance that a predicate assigns to each of its attributes under observation, w_j^i, stands for the relevance of a_j for the predicate p_i and it is given by $V_i(X) = \sum w_i j V^i j(x)$, for all p_i. On the other hand, the predicate scoring function, when associated to a value $x = (x_1, ..., x_n)$ in a multi-dimensional space defined by the attribute domains, is given in the form $V_i(X) = \sum w_i j V^i j(x)$.

Therefore, it is now possible to measure the quality-of-information that stems from a continuous logic program, by posting Qi values into a multi- dimensional space, whose axes denote the program predicates with a numbering ranging from 0 (at the center) to 1. The area delimited by the arcs gives a measure of the quality-of-information carried out by each problem solution that may be under consideration, therefore defining the process of quantification of the morally preferable relation, as it is stated above in formal terms.

4.4.4 Model Behavior

Case 1 Mr. PD is a man with 81 years, a long background of cardiopathy and diabetes is admitted in an ICU with fever, hypertension and dyspnea. The thorax radiography is compatible with Acute Respiratory Distress Syndrome (ARDS) and the arterial partial oxygen tension (PaO2) is of 50 mmHg. This condition is often fatal, usually requiring mechanical ventilation and although the short-time mortality in these cases has been decreasing, the probability of mortality is considerably high and moreover this procedure results in a low quality-adjusted survival in the first year after ARDS [8, 13]. At the noon service meeting, while analyzing the current cases, the assistant physician asks the interns whether in light of the survival rates, treatment costs and probable low quality of life, should the ICU resources be used with this 81 years old men.

Case 2

During this meeting Mrs. GB, a woman with 36 years interned at the same hospital due to a car accident and diagnosed with sepsis, Acute Lung Injury (ALI) and Glasgow coma scale of 3, shows breathing complications and needs to be admitted an ICU. The level of its PaO2 and the severity of the ALI indicated a pressing need for mechanical ventilation and intensive care. However the number of beds in the ICU is limited and for this matter Mr. PD would have to be changed to another service. Due to the fragile state of Mr. PD this procedure is problematical, but considering his clinical status, complications and age with Mrs. GB, the greater probability of her to full recover with better quality of life tends to tip the balance from a critical point of view. In light of this context, how should the assistant physician act?

The continuous logic program for predicate survival-rate:

$\neg\, survival - rate(X,Y) \leftarrow notsurvival - rate(X,Y) and notexceptionsurvival - rate(X,Y),$

$exceptionsurvival - rate(X,Y) \leftarrow survival - rate(X, unknown - survival - rate),$

$survival - rate(X,Y) \leftarrow ards(X) and pao2(X, low) and evaluate(X,Y),$

$exceptionsurvival - rate(gb, 0.5),$

$?((exceptionsurvival - rate(X,Y) or exceptionsurvival - rate(X,Z)) and$

$\neg\,(exceptionsurvival - rate(X,Y) and exceptionsurvival - rate(X,Z))$

/This invariant states that the exceptions to the predicate survival-rate follow an exclusive or/

$agsurvival - rate$

The continuous logic program for predicate survival-quality:

$\neg\ survival-quality(X,Y) \leftarrow not\ survival-quality(X,Y)$
$and\ not\ exceptionsurvival-quality(X,Y),$
$exceptionsurvival-quality(X,Y) \leftarrow survival-rate(X,unknown-survival-quality),$
$survival-quality(gb,0.8),\ exceptionsurvival-quality(pd,0.1),\ ?((exceptionsurvival-quality(X,Y) orexceptionsurvival-quality(X,Z))and$
$\neg(exceptionsurvival-quality(X,Y)andexceptionsurvival-quality(X,Z))\ agsurvival-quality$

The continous logic program for predicate cost:

$\neg cost(X,Y) \leftarrow notcost(X,Y)andnotexceptioncost(X,Y),$
$exceptioncost(X,Y) \leftarrow cost(X,unknown-cost),$
$cost(gb,unknown-cost),$
$cost(pd,unknown-cost),$
$?((exceptioncost(X,Y)orexceptioncost(X,Z))and$
$\neg(exceptioncost(X,Y)andexceptioncost(X,Z))$
$agcost$

In this specific case we assume that costs are unknown, so they will be considered as null values for the calculi.

4.4.5 Discussion

Several predicates have to generated and considered in the CLP construction, in order to demonstrate theorems for decision taking. This is a bi-directional process because beyond the organizational, functional, technical and scientific requisites, one may have to attend ethical and legal ones, as well as data quality, information security, access control and privacy. This generation is made from the nosocomial Electronic Health Records (EHR). EHR is a core application which covers horizontally the health care unit and makes possible a transverse analysis of medical records along the several services, units or treated pathologies, bringing to healthcare units new computational models, technologies and tools, based on data warehouses, agents, multi-agent systems and ambient intelligence. An EHR is an assembly of standardized documents, ordered and concise, directed to the register of actions and medical procedures; a set of information compiled by physicians and others health professionals; a register of integral facts, containing all the information regarding patient health data; and a follow up of the risk values and clinical profile. The main goal is to replace hard documents by electronic ones, increasing data processing and reducing time and costs. The patient assistance will be more effective, faster and quality will be improved.

Whatever form of an information society related to healthcare we can imagine, it will be based on three basic components, namely raw medical data, reconstructed medical data and derived medical data. Indeed, clinical research and practice involve a process to collect data to systematize knowledge about patients, their health status and the motives of the health care admittance. At the same time, data has to be registered in a structured and organized way, making effective automation and supporting using Information Technologies. For example, from an information repository, one may have collected patient data, which are registered in an efficient, consistent, clear and structured way to improve disease knowledge and therapy; the medical processes for registering data are complemented with the information interchange between the different physicians that work around the patient; and the clinical data recording are guaranteed in the EHR application and procedural context. Interoperability will allow for sharing information among several information systems.

The process to collect data comes from Problem Oriented Medical Record (POMR) method. This is a format for clinical recording consisting of a problem list; a database including the patient history with physical examination and clinical findings; diagnostic, therapeutic and educational plans; and a daily SOAP (Subjective, Objective, Assessment and Plan) progress note. The problem list serves as an index for the reader, each problem being followed through until resolution. This system widely influences note keeping by recognizing the four different phases of the decision making process: data collection; formulation of problems; and devising a management plan; and reviewing the situation and revising the plan if necessary.

4.4.6 Conclusion

Different methodologies based on artificial intelligence have been proposed to model ethical reasoning, however we consider that continous logic programming expresses characteristics that overcome the main shortcomings of other techniques such as black-box techniques. One of the main advantages of using CLP concern the context of ethical modeling itself, as most of the trustworthy knowledge is based on deontological principles and is oriented towards experts consideration. The principle and exception modeling demonstrated presents a modeling clearly understandable by experts, traceable through proof trees and which processing is clearly identifiable, predictable and updatable.

The ultimate goal of, using CLP is not to simulate moral reasoning itself, but rather enable decision support architectures, which take into account moral context. That is the reason why the possibility to justify moral decision and doubt on real-time to clinical staff is of the essence. Using such modeling principles, this staff could recur to moral decision support on real time and understand the line of reasoning implicit in the decision advised by the system. Although a long path must be walked before such moral aware decision support system to be implemented, this study of moral modeling and representation is of the essence to set the basilar structure in which morality can be imbedded in future systems.

References

Anderson M, Anderson SL, Armen C (2006) An approach to computing ethics. Ieee Intelligent Systems 21(4):56–63, iSI Document Delivery No.: 068PE Times Cited: 1 Cited Reference Count: 11 Anderson, Michael Anderson, Susan Leigh Armen, Chris Ieee computer soc Los alamitos

Danbury CM, Waldmann CS (2006) Ethics and law in the intensive care unit. Best Practice & Research Clinical Anaesthesiology 20(4):589–603

Floares AG (2007) A reverse engineering algorithm for neural networks, applied to the subthalamopallidal network of basal ganglia. Neural Networks 21(2-3):379–386

Guarini M (2006) Particularism and the classification and reclassification of moral cases. IEEE Intell Syst 21(4):7

Horty JF (1994) Moral dilemmas and nonmonotonic logic. Journal of Philosophical Logic 23(1):35–65, iSI Document Delivery No.: MX374 Times Cited: 25 Cited Reference Count: 28 Kluwer academic publ Dordrecht

Jonsen AR, Siegler M, Winslade WJ (1997) Clinical Ethics, 4th edn. McGraw-Hill

Machado J, Miranda MFM, Abelha A, Neves J, Neves J (2009) Modeling medical ethics through intelligent agents

Nugent C, Cunningham P (2005) A case-based explanation system for black-box systems. Artificial Intelligence Review 24(2):163–178

Powers TM (2006) Prospects for a kantian machine. IEEE Intelligent Systems 21(4):46–51, iSI Document Delivery No.: 068PE Times Cited: 2 Cited Reference Count: 9 Powers, Thomas M. Ieee computer soc Los alamitos

Serrão D, Nunes R (1998) Ética em Cuidados de Saúde. Porto Editora

Sprung CL, Cohen SL, Sjokvist P, Baras M, Bulow HH, Hovilehto S, Ledoux D, Lippert A, Maia P, Phelan D, Schobersberger W, Wennberg E, Woodcock T (2003) End-of-life practices in european intensive care units: The ethicus study. Journal Of the American Medical Association 290(6):790–797

Tonkens R (2009) A challenge for machine ethics. Minds and Machines 19(3):421–438, iSI Document Delivery No.: 519FS Times Cited: 0 Cited Reference Count: 32 Tonkens, Ryan Springer Dordrecht

Chapter 5
Conclusion

5.1 Discussion of results and conclusions

The core objective and aimed result of this research was always the resolution of existing problems found in healthcare interoperability. Henceforth, associated to its theoretical content, this research possessed a deeply strong practical counterpart. For this purpose, the challenges and feedback from the dynamic and complex production environment in the participant healthcare institutions was essential to improve and devise adaptive models towards a better understanding on how implement interoperability.

The proposed archetype addressed difficulties not only that were clear from the literature review, but also issues found in the specificities of the portuguese healthcare system in terms of technology information. One of the most significant research opportunities risen from the fact that the participant institutions, all used software from distinct companies either for transversal services (horizontal dependency) or department oriented. Regardless of their role and broadness in the healthcare, these systems all had to gain to be connected with one another specially due to the collaborative perspective on medical practice. Not only towards ease of use from the user but also for diminishing the impact of applications isolation behaviour, integration and interoperability are a great asset for healthcare information systems. Moreover, from the perspective of a broaden Health Information System as a synergy of applications rather than a core monolithic system, can only be implemented with an high level of interoperability.

Although the proposal of a unique agency to consolidate the communication among systems points towards the ideals of a centralization model, the usage of a modular approach and the agent paradigm allow to overcome many of the limitations associated to such models.

Even though the development of software for production purposes when based on a thesis research is not very common, this cooperation with the "Centro Hospitalar do Porto" and the "Centro Hopitalar do Alto Ave" was very successful from a synergitical perspective. The resulting archetype and its implementation so solve practical issues not only enriched this research with a pragmatical and real world difficulties, as it also allowed these institutions to improve their health information systems performance and the quality of the information provided.

Quality of information is of the essence not only for practitioners to provide a better care of patients but also for technicians, administrators and other staff in order to optimize processes, diminish waste and prevent errors. In fact, the ability of the practitioners to provide a correct diagnosis and better care all together can directly depend from the quality of the information provided. Similarly the internal procedures for auxiliary exams, prescriptions and alike are not only important to medical staff but also to technicians who perform them and administrators that have to make business decisions and evaluations in light of this information. Henceforth, the performance and quality of service of the healthcare institution appears clearly dependent of the manner information flows within the institution.

This cornerstone role of interoperability in healthcare allowed for several opportunities to enrich the model proposed in this thesis. One of the major difficulties in heterogeneous systems that are part of a complex environment, is the creation of information mismatch among them, either by incoherence or lack of communication all together. Therefore the proposed model is based on a aggregation of information in a coherent base of knowledge, that acts as a master and distributable source of information among systems. This allows a repository for all information within the institution as well as important stepping stone for a more complete and structured electronic health record centered in the patient, as was proven in the CHP hospital centre. The presence of this iteration system within the hospital allowed the development of an EHR that aggregates information from all systems within the hospital center. This was indeed a strong deliverable from this research and an inciting advantage of consolidation of information and integration.

However, the centralization of information in a unique system is clearly a point of failure for the whole infrastructure. This limitation was addressed using the multi-agent system allied with service oriented development and a distributed databases. The implementation results indicated that the approach used by the archetype addressed this issues not only in a theoretical perspective but also from the prototype's implementation point of view.

Regarding the use of intelligent agent-based systems in moral complex environments in this thesis it is proposed and demonstrated a solid model for ethical reasoning. It defends that continuous logic programming expresses characteristics that overcome the main shortcomings of other techniques such as black-box techniques or probabilistic algorithms. It also adapts strongly to the moral reasoning of human agents in ethical situations, as is verified by the fact thats most of the trustworthy knowledge is based on deontological principles and is oriented towards experts consideration. Moreover, the principle and exception modeling demonstrated is a clear and perceivable manner to be understandable by human experts as well as being traceable through proof trees and which processing is clearly identifiable, predictable and updatable.

Despite the fact that this is not an ultimate solution for moral aware decision support system to be implemented, this study of moral modeling and representation is a cornerstone in oder to further understand and define the organization and manner in which morality can be imbedded in future systems.

Most of the the results from this thesis and underlying research were of course published in order to disseminate the knowledge resulting from them. These studies were submitted for a peer review process and accepted by the scientific community as valuable research work that deserved being shared among the scientific community. Their acceptance and the visibility of the publications itself asserts for the quality of the developed research and allows to conclude that the proposed archetype and the novel application of existing paradigms gained reconnaissance due to this work.

5.2 Summary of research deliverables and achievements

The results of the research from this thesis can be defined as the following list of deliverables and achievements:

- models and underlying proposed solutions for interoperability in healthcare based on new paradigms and technologies
- technical description such models in functional and technological perspectives
- development of a software basic platform from which the proposed archetype can be implemented adaptable to each environment
- implementation of several prototypes in production healthcare environment
- implementation of case studies that were documented and published on conferences and peer review journals
- study of the implementation of such systems an prototypes regarding the implementation of
- model for ethical reasoning for agents in healthcare
- several publications in book chapters and proceedings with relevance in the research area

5.3 Associated publications

Resulting directly or by association from the research performed this thesis there were several publications in journals, as book chapters and articles from procedings.

5.3.1 Journals

- Miguel Miranda, Maria Salazar, Filipe Portela, Manuel Santos, António Abelha, José Neves and José Machado, Multi-agent systems for HL7 interoperability services, Procedia Technology, Volume 5, Elsevier, 2012.

5.3.2 Book Chapters

- Miguel Miranda, José Machado, António Abelha and José Neves, Healthcare interoperability through a JADE based multi-agent platform, INTELLIGENT DISTRIBUTED COMPUTING VI, Studies in Computational Intelligence, Volume 446/2013, Springer (2013).
  Classified as a Book Chapter by Springer.
- Rui Rodrigues, Pedro Gonçalves, Miguel Miranda, Carlos Portela, Manuel Santos, José Neves, António Abelha and José Machado, An Intelligent Patient Monitoring System, 20th Internation Symposium on Methodologies for Intelligent Systems, 2012 World Intelligence Congress, Macau (LNCS Springer) (2012).
  Classified as a Book Chapter by Springer.
- Miguel Miranda, Gabriel Pontes, Pedro Gonçalves,Hugo Peixoto, Manuel Santos, Antonio Abelha, José Machado: Modelling Intelligent Behaviours in Multi-agent Based HL7, Computer and Information Science 2010 - Studies in Computational Intelligence, Services, vol. 317, pp. 95. Springer Berlin / Heidelberg (2010).
  Classified as a Book Chapter by Springer.
- Miguel Miranda, José Machado, António Abelha, José Neves and João Neves, Evolutionary Intelligence in Agent Modeling and Interoperability, Ambient Intelligence - Software and Applications, Advances in Intelligent and Soft Computing, Volume 92/2011, Springer. (2011).
  Classified as a Book Chapter by Springer.
- Miguel Miranda, José Machado, António Abelha, Gabriel Pontes and José Neves: A Step Towards Medical Ethics Modeling, E-Health - IFIP Advances in Information and Communication Technology, vol. 335, pp. 27-36, Springer Boston (2010).
  Classified as a Book Chapter by Springer.
- José Machado, Miguel Miranda, Gabriel Pontes, António Abelha and José Neves: Morality in Group Decision Support Systems in Medicine, Intelligent Distributed Computing IV - Studies in Computational Intelligence, vol. 315, pp. 191-200, Springer Berlin / Heidelberg (2010).
  Classified as a Book Chapter by Springer.
- José Machado, Miguel Miranda, António Abelha, José Neves and João Neves: Modeling Medical Ethics through Intelligent Agents, Software Services for e-Business and e-Society - IFIP Advances in Information and Communication Technology, vol. 305, pp. 112-122, Springer Boston

(2009).
Classified as a Book Chapter by Springer.

- Miguel Miranda, António Abelha, Manuel Santos, José Machado and José Neves, A Group decision support system for staging of cancer: Electronic Healthcare, Dasun Weerasinghe (ed.), Springer-Verlag, Series Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp 114-121, ISBN 978-3-642-00412-4, (Revised Selected Papers of The 1st International Conference on Electronic Healthcare in The 21st Century, City University, London, England, in September 8 - 9, 2008).
Classified as a Book Chapter by Springer.

5.3.3 Procedings

- Miguel Miranda, Gabriel Pontes, António Abelha, José Neves and José Machado, Agent Based Interoperability in Hospital Information Systems, 5th IEEE International Conference on Biomedical Engineering and Informatics. Chongqing, China, 2012.
- Rui Rodrigues, Pedro Gonçalves, Miguel Miranda, Carlos Portela, Manuel Santos, José Neves, António Abelha and José Machado, Monitoring Intelligent System for the Intensive Care Unit using RFID and Multi-Agent Systems, 4ht IEEE International Conference on Industrial Engineering and Engineering Management.
- Gabriel Pontes, Ana Duarte, David Cuevas, Maria Salazar, Miguel Miranda, António Abelha and José Machado, A Moral Decision Support System in Medicine, European Simulation and Modelling Conference (ESM 2011), Guimarães, Portugal, EUROPEAN SIMULATION AND MODELLING CONFERENCE 2011.
- Miguel Miranda, Júlio Duarte, António Abelha, José Machado, José Neves and João Neves, Interoperabity in Healthcare, in Proceedings of the ESM 2010, Hasselt, Belgium.
To be indexed at the Web of Science
- José Machado, Miguel Miranda, Pedro Gonçalves, António Abelha, José Neves and Alberto Marques, AIDATrace - Interoperation Platform for Active Monitoring in Healthcare Environments, ISC 2010, Budapeste, Hungria.
To be indexed at the Web of Science
- Miguel Miranda, Manuel Santos, António Abelha, José Machado and José Neves, An Agent-based Architecture for Cancer Staging, in Proceedings of 11th International Conference on Enterprise Information Systems (ICEIS 2009), Milan, Italy, 2009.
Indexed at the Web of Science
- David Belo, Miguel Miranda, António Abelha, José Machado and José Neves, Quality in Transplantation Service using Electronic Medical Record, in Proceedings of the 7th annual Indus-

trial Simulation Conference, Loughborough, UK, 2009.
Indexed at the Web of Science

- Julio Duarte, Miguel Miranda, António Abelha, José Machado, José Neves, Manuel Santos, Carlos Alberto, Maria Salazar, César Quintas, Alexandra Ferreira, Joao Neves, Agent-based Group Decision Support in Medicine, in Proceedings of Worldcomp 2009, 2009 International Conference on Artifcial Intelligence, Las Vegas, USA, 2009.
  Indexed at DBLP
- Machado M., Miranda M., Pontes G., Santos D., Santos M., Abelha A., Neves J., Intelligent Agents and Medical Moral Dilemmas, in Proceedings of the 8th WSEAS International Conference on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '09), Hangzhou, China, May 20-22, 2009.
  To be indexed at the Web of Science
- Miguel Miranda, Júlio Duarte, António Abelha, José Machado and José Neves, Interoperability and Healthcare, in Proceedings of ESM 2009, 23rd annual European Simulation and Modelling Conference, Leicester, United Kingdom, 2009.
  Indexed at the Web of Science

5.4 Future work

Research work is never ended, it's aim is always to shed some light into previously undiscovered or unproven theories, to challenge existing paradigms and propose new extraordinary ones. Henceforth, this thesis is only but a stepping stone for the use of cutting-edge and novel technologies and methodologies towards improving the level of interoperability achieved from this archetype.

There are strong results encouraging and adding a strong impulse towards the use of paradigms deriving from artificial intelligence in order to solve interoperability problems. The proposed archetype is far from a single true answer towards interoperability. It is however a proof that artificial intelligences and its fields like agent-systems or knowledge representation have much to offer as possible tools to increase the level of interoperability in complex heterogeneous systems.

References

ACM (2009) Code of ethics and professional conduct. Tech. rep., ACM, URL http://www.acm.org/about/code-of-ethics

Andrade F, Neves J, Novais P, Machado J (2004) Software agents as legal persons. In: Camarinha-Matos LM (ed) Virtual Enterprises and Collaborative Networks, Kluwer, pp 123–132

Andrade F, Neves J, Novais P, Machado J, Abelha A (2005) Legal security and credibility in agent based virtual enterprises. In: Camarinha-Matos L, Afsarmanesh H, Ortiz A (eds) Collaborative Networks and Their Breeding Environments, IFIP — The International Federation for Information Processing, vol 186, Springer US, pp 503–512

Bauer C, King G (2006) Java Persistence with Hibernate. Manning Publications Co., Greenwich, CT, USA

Bellifemine F, Caire G, Greenwood D (2007) Developing Multi-Agent Systems with JADE. John Wiley & Sons, ISBN: 978-0470057476

Bellifemine F, Caire G, Trucco T, Rimassa G (2010a) Jade programmer's guide. Tech. rep., Telecom Italia S.p.A.

Bellifemine F, Caire G, Trucco T, Rimassa G, Mungenast R (2010b) Jade administrator's guide. Tech. rep., JADE Board

Berg M (2004) Health Information Management: Integrating Information Technology in Health Care Work. Routledge

Blobel B (2006) Advanced and secure architectural ehr approaches. International Journal of Medical Informatics 75(3-4):185 – 190, DOI 10.1016/j.ijmedinf.2005.07.017

Bodart C, Lippeveld T, Sauerborn R (eds) (2000) Design and Implementation of Health Information Systems. World Health Organization

Bryson JJ (2000) Cross-paradigm analysis of autonomous agent architecture. Journal of Experimental and Theoretical Artificial Intelligence 12(2):165–190

Carr CD, Moore SM (2003) Ihe: a model for driving adoption of standards. Computerized Medical Imaging and Graphics 27(2-3):137 – 146, DOI 10.1016/S0895-6111(02)00087-3

Cavalcante RC, Bittencourt II, da Silva AP, Silva M, Costa E, Santos R (2012) A survey of security in multi-agent systems. Expert Systems with Applications 39(5):4835 – 4846

Cranefield S, Purvis MK (1996) An agent-based architecture for software tool coordination. In: Cavedon L, Rao AS, Wobcke W (eds) PRICAI Workshop on Intelligent Agent Systems, Springer, Lecture Notes in Computer Science, vol 1209, pp 44–58

Deigh J (1992) Ethics and Personality: Essays in Moral Psychology. University of Chicago Press

Deigh J (1996) The Sources of Moral Agency: Essays in Moral Psychology and Freudian Theory. Cambridge University Press

Dreiseitl S, Binder M (2005) Do physicians value decision support? a look at the effect of decision support systems on physician opinion. Artificial Intelligence in Medicine 33(1):25 – 30, DOI 10.1016/j.artmed.2004.07.007

Duarte J, Miranda MFM, Abelha A, Santos M, Machado J, Neves J, Alberto C, Salazar M, Quintas C, Ferreira AMSF, Neves J (2009) Agent-based group decision support in medicine. In: Arabnia HR, de la Fuente D, Olivas JA (eds) IC-AI, CSREA Press, pp 115–121

Dupuits FMHM, Hasman A (1995) Evaluation of a decision support system in a medical environment. International Journal of Bio-Medical Computing 38(2):155 – 165, DOI 10.1016/0020-7101(94)01047-5

Fischer K (1993) The rule-based multi-agent system magsy. In: Keele University

Fischer K, Fischer K (1993) Verteiltes und kooperatives Planen in einer flexiblen Fertigungsumgebung. Dissertationen zur künstlichen Intelligenz, Infix

Floridi L, Sanders JW (2004) On the morality of artificial agents. Minds Mach 4(3):349–679

Georgeff MP, Pell B, Pollack ME, Tambe M, Wooldridge M (1999) The belief-desire-intention model of agency. In: Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages, Springer-Verlag, London, UK, UK, ATAL '98, pp 1–10

Haux R (2006) Individualization, globalization and health - about sustainable information technologies and the aim of medical informatics. International Journal of Medical Informatics 75(12):795 – 808, jochen Moehr Special Issue

Himma KE (2008) Artificial agency, consciousness, and the criteria for moral agency: what properties must an artificial agent have to be a moral agent? In: Ethics and Information Technology, Springer

Initiative DCM (2010) Interoperability levels for dublin core metadata. Tech. rep., DCMI, URL http://dublincore.org/documents/interoperability-levels/

ISO/TC215 (2005) Health informatics profiling framework. Tech. Rep. ISO/TR 17119:2005, ISO

Jennings NR, Sycara K, Wooldridge M (1998) A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems 1:7–38, URL http://dx.doi.org/10.1023/A:1010090405266, 10.1023/A:1010090405266

Kirsh W (ed) (2008) Encyclopedia of Public Health, vol 1. Springer Science

Lopez DM, Blobel BG (2009) A development framework for semantically interoperable health information systems. International Journal of Medical Informatics 78(2):83 – 103, DOI 10.1016/j.ijmedinf.2008.05.009

Machado J, Abelha A, Novais P, Neves J, Neves J (2008) Improving patient assistance and medical practices through intelligent agents. In: Workshop on Health Informatics, AAMAS

Machado J, Miranda M, Abelha A, Neves J, Neves J (2009a) Modeling medical ethics through intelligent agents. Software Services for e-Business and e-Society pp 112–122

Machado J, Miranda M, Pontes G, dos Santos D, Santos M, Abelha A, Neves J (2009b) Intelligent agents and medical moral dilemmas

Marreiros G, Santos R, Novais P, Machado J, Ramos C, Neves J, Bula-Cruz J (2007) Argumentation-based decision making in ambient intelligence environments. Progress in Artificial Intelligence pp 309–322

Minter D, Linwood J (2005) Pro Hibernate 3. Apresspod Series, Apress, URL http://books.google.pt/books?id=AM1Od544ei0C

Minter D, Linwood J (2006) Beginning Hibernate: From Novice to Professional. Beginning: from Novice to Professional, Apress

Miranda M, Abelha A, Santos M, Machado J, Neves J (2009a) A group decision support system for staging of cancer. Electronic Healthcare pp 114–121

Miranda M, Duarte J, Abelha A, Machado J, Neves J (2009b) Interoperability and healthcare. In: European Simulation and Modelling Conference 09

Mora M, Lopes J, Viccariz R, Coelho H (1999) Bdi models and systems: Reducing the gap. In: Müller J, Rao A, Singh M (eds) Intelligent Agents V: Agents Theories, Architectures, and Languages, Lecture Notes in Computer Science, vol 1555, Springer Berlin Heidelberg, pp 11–27

Müller J (1997) Control architectures for autonomous and interacting agents: A survey. In: Cavedon L, Rao A, Wobcke W (eds) Intelligent Agent Systems Theoretical and Practical Issues, Lecture Notes in Computer Science, vol 1209, Springer Berlin / Heidelberg, pp 1–26

Mykkänen J, Tuomainen M (2008) An evaluation and selection framework for interoperability standards. Information and Software Technology 50(3):176 – 197

Newell A, Simon HA (1976) Computer science as empirical inquiry: symbols and search. Commun ACM 19(3):113–126, DOI 10.1145/360018.360022, URL http://doi.acm.org/10.1145/360018.360022

Nwana HS (1996) Software agents: An overview. In: Knowledge Engineering Review, Cambridge University Press, vol 11, pp 1–40

Ohe K, Kaihara S (1996) Implementation of hl7 to client-server hospital information system (his) in the university of tokyo hospital. Journal of Medical Systems 20(4):197–205, URL http://dx.doi.org/10.1007/BF02263391

Padgham L, Lambrix P (2005) Formalisations of capabilities for bdi-agents. Autonomous Agents and Multi-Agent Systems 10:249–271, 10.1007/s10458-004-4345-2

Pereira LM, Saptawijaya A (2007) Modelling morality with prospective logic. In: Procs. 13th Portuguese Intl.Conf. on Artificial Intelligence (EPIA'07), LNAI, Springer

PHII (2004) Guiding principles for effective health information systems. Public Health Informatics Institute, Decatur Institute, GA

Rao A, Georgeff MP (1995) Bdi agents: from theory to practice. In: Proceedings of the First International Conference on Multi Agent Systems, ICMAS'95, MIT Press, San Francisco EUA

Talukdar S (2004) Multi-agent systems. In: Power Engineering Society General Meeting, 2004. IEEE, pp 59 – 60 Vol.1, DOI 10.1109/PES.2004.1372753

Tolk DA, Muguira JA (2003) The levels of conceptual interoperability model. In: in '2003 Fall Simulation Interoperability Workshop

Vernadat FB (2010) Technical, semantic and organizational issues of enterprise interoperability and networking. Annual Reviews in Control 34(1):139–144

Williams M, Hunter A (2007) Harnessing ontologies for argument-based decision-making in breast cancer. In: ICTAI '07: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, Washington, DC, USA, pp 254–261

Wooldridge M (1997) Issues in agent-based software engineering. In: Kandzia P, Klusch M (eds) Cooperative Information Agents, Lecture Notes in Computer Science, vol 1202, Springer Berlin / Heidelberg, pp 1–18

Wooldridge M (2009) An Introduction to MultiAgent Systems, 2nd edn. Wiley Publishing

Wooldridge M, Jennings N (1995a) Agent theories, architectures, and languages: A survey. In: Wooldridge M, Jennings N (eds) Intelligent Agents, Lecture Notes in Computer Science, vol 890, Springer Berlin / Heidelberg, pp 1–39

Wooldridge M, Jennings NR (1995b) Intelligent agents: Theory and practice. Knowledge Engineering Review 10:115–152

Wooldridge MJ (1999) Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence, MIT Press, chap 1

Zaleski J (2009) Integrating Device Data into the Electronic Medical Record: A Developer's Guide to Design and a Practitioner's Guide to Application, 1st edn. Wiley-VCH

Zhang J, Xu X, Hong L, Yan Y (2012) Consensus recovery of multi-agent systems subjected to failures. International Journal of Control 85(3):280–286, DOI 10.1080/00207179.2011.646313

Appendix A

Appendix

A.1 Tables

Table A.1: Hibernate Configuration Property Names and Descriptions as described in Minter and Linwood (2005)

Property Name	Description
hibernate.connection.provider_class	Class that implements Hibernate's ConnectionProvider interface.
hibernate.connection.driver_class	The JDBC driver class.
hibernate.connection.isolation	The transaction isolation level for the JDBC connection.
hibernate.connection.url	The JDBC URL to the database instance.
hibernate.connection.username	Database username.
hibernate.connection.password	Database password.
hibernate.connection.autocommit	Uses autocommit for the JDBC connection.
hibernate.connection.pool_size	Limits the number of connections waiting in the Hibernate database connection pool.
hibernate.connection.datasource	Datasource name for a container-managed data source.
hibernate.connection.<JDBCpropertyname>	Passes any JDBC property you would like to the JDBC connection—for instance, hibernate.connection.debuglevel=info would pass a JDBC property called debuglevel.
hibernate.jndi.class	Initial context class for JNDI.
hibernate.jndi.url	Provides URL for JNDI.
hibernate.jndi.<JNDIpropertyname>	Passes any JNDI property you would like to the JNDI InitialContext.

hibernate.session_factory_name	If this property is set, the Hibernate session factory will bind to this JNDI name.
hibernate.dialect	SQL dialect to use for Hibernate, varies by database. See section on SQL dialects.
hibernate.default_schema	Default database owner name that Hibernate uses to generate SQL for unqualified table names.
hibernate.default_catalog	Default database catalog name that Hibernate uses to generate SQL for unqualified table names.
hibernate.show_sql	Logs the generated SQL commands. Generates SQL with comments.
hibernate.use_sql_comments	Generates SQL with comments.
hibernate.max_fetch_depth	Determines how deep Hibernate will go to fetch the results of an outer join. Used by Hibernate's outer join loader.
hibernate.jdbc.use_streams_for_binary	Determines if binary data is read or written over JDBC as streams.
hibernate.jdbc.use_scrollable_resultset	Determines if Hibernate will use JDBC scrollable resultsets for a user-provided JDBC connection.
hibernate.jdbc.use_get_generated_keys	If the database driver supports the JDBC 3 autogenerated keys API, Hibernate will retrieve any generated keys from the statement after it executes a SQL query.
hibernate.jdbc.fetch_size	Determines how many rows the JDBC connection will try and buffer with every fetch. This is a balance between memory and minimizing database network traffic.
hibernate.jdbc.batch_size	The maximum batch size for updates.
hibernate.jdbc.factory_class	The class name of a custom implementation of the org.hibernate.jdbc.Batcher interface for controlling JDBC prepared statements.

hibernate.jdbc.batch_versioned_data	Determines if Hibernate batches versioned data, which will depend on your JDBC driver properly implementing row counts for batch updates. Hibernate uses the row count to determine if the update was successful.
hibernate.xml.output_stylesheet	Specifies an XSLT stylesheet for Hibernate's XML data binder. Requires xalan.jar.
hibernate.c3p0.max_size	Maximum size of the connection pool for C3PO.
hibernate.c3p0.min_size	Minimum size of the connection pool for C3PO.
hibernate.c3p0.timeout	Timeout for C3PO (in seconds).
hibernate.c3p0.max_statements	Upper limit for the SQL statement cache for C3PO.
hibernate.c3p0.acquire_increment	After connection pool is completely utilized, determines how many new connections are added to the pool.
hibernate.c3p0.idle_test_period	Determines how long to wait before a connection is validated.
hibernate.proxool	Prefix for the Proxool database connection pool.
hibernate.proxool.xml	Path to a Proxool XML configuration file.
hibernate.proxool.properties	Path to a Proxool properties file.
hibernate.proxool.existing_pool	Configures Proxool with an existing pool.
hibernate.proxool.pool_alias	Alias to use for any of the above configured Proxool pools.
hibernate.transaction.auto_close_session	Closes session automatically after a transaction.
hibernate.transaction.flush_before_completion	Automatically flushes before completion.
hibernate.transaction.factory_class	Specifies a class that implements the org.hibernate.transaction. TransactionFactory interface.
hibernate.transaction.manager_lookup_class	Specifies a class that implements the org.hibernate.transaction. TransactionManagerLookup interface.
jta.UserTransaction	The JNDI name for the UserTransaction object.

hibernate.cache.provider_class	Specifies a class that implements the org.hibernate.cache.CacheProvider interface.
hibernate.cache.use_query_cache	Specifies whether or not to use the query cache.
hibernate.cache.query_cache_factory	Specifies a class that implements the org.hibernate.cache.QueryCacheFactory interface for getting QueryCache objects.
hibernate.cache.use_second_level_cache	Determines whether or not to use the Hibernate second level cache.
hibernate.cache.use_minimal_puts	Configures the cache to favor minimal puts over minimal gets.
hibernate.cache.region_prefix	The prefix to use for the name of the cache.
hibernate.generate_statistics	Determines whether statistics are collected.
hibernate.use_identifier_rollback	Determines whether Hibernate uses identifier rollback.
hibernate.cglib.use_reflection_optimizer	Instead of using slower standard Java reflection, uses the CGLib code generation library to optimize access to business object properties. Slower at startup if this is enabled, but faster runtime performance.
hibernate.query.factory_class	Specifies an HQL query factory class name.
hibernate.query.substitutions	Any possible SQL token substitutions Hibernate should use.
hibernate.hbm2ddl.auto	Automatically creates, updates, or drops database schema on startup and shut down. There are three possible values: create, create- drop, and update. Be careful with create-drop!
hibernate.sql_exception_converter	Specifies which SQLExceptionConverter to use to convert SQLExceptions into JDBCExceptions.
hibernate.wrap_result_sets	Turns on JDBC result set wrapping with column names.
hibernate.order_updates	Order SQL update statements by each primary key.

Table A.2: Hibernate Cache Providers

Cache	Provider Class	Type
Hashtable	org.hibernate.cache.HashtableCacheProvider	memory
EHCache	org.hibernate.cache.EhCacheProvider	memory, disk
OSCache	org.hibernate.cache.OSCacheProvider	memory, disk
SwarmCache	org.hibernate.cache.SwarmCacheProvider	clustered
JBoss Cache 1.x	org.hibernate.cache.TreeCacheProvider	clustered, transactional
JBoss Cache 2	org.hibernate.cache.jbc.JBossCacheRegionFactory	clustered, transactional

A.2 Listings

Listing A.1: Hibernate configuration file for the database details.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
   Configuration DTD 3.0//EN" "http://www.hibernate.org/dtd/hibernate-
   configuration-3.0.dtd">
<hibernate-configuration>
      <session-factory>
            <property name="hibernate.dialect">org.hibernate.dialect.
               Oracle10gDialect</property>
            <property name="hibernate.connection.driver_class">oracle.jdbc.
               OracleDriver</property>
            <property name="hibernate.connection.url">jdbc:oracle:thin:@chp
               -ora01:1521:aida1</property>
            <property name="hibernate.connection.username">*****</property>
            <property name="hibernate.connection.password">*********</
               property>
            <!-- Enable autocommit -->
            <property name="hibernate.connection.autocommit">true</property
               >
            <!-- Enable the JDBC to try CLOB insertion with strings over 4k
                -->
            <property name="SetBigStringTryClob">true</property>
            <!-- Set Hibernate's number of operations to batch before
               flushing to database -->
            <property name="hibernate.jdbc.batch_size">0</property>
```

```
<!-- Enable Hibernate's automatic session context management --
   >
<property name="current_session_context_class">thread</property
   >
<!-- Disable the second-level cache -->
<property name="cache.provider_class">org.hibernate.cache.
   NoCacheProvider</property>
<!-- Echo all executed SQL to stdout -->
<property name="show_sql">false</property>
<!-- Connection Pooling -->
<!-- configuration pool via c3p0 -->
<property name="hibernate.c3p0.acquire_increment">2</property>
<property name="hibernate.c3p0.idle_test_period">100</property>
<!-- seconds -->
<property name="hibernate.c3p0.max_size">100</property>
<property name="hibernate.c3p0.max_statements">20</property>
<property name="hibernate.c3p0.min_size">1</property>
<property name="hibernate.c3p0.timeout">100</property>
<!-- seconds -->
<!-- MAPPINGS -->
<mapping resource="aida/database/hibernate/sil/schema/PdPedidos
   .hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/
   EpisodiosCriadosSonho.hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/Dpedidos.
   hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/
   PdPedLinhas.hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/
   EpisodiosAida.hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/AgdProt.
   hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/Protigif.
   hbm.xml"/>
<mapping resource="aida/database/hibernate/sil/schema/Protocolo
   .hbm.xml"/>
```

```
        <mapping resource="aida/database/hibernate/sil/schema/Gpedproto
            .hbm.xml"/>
        <mapping resource="aida/database/hibernate/sil/schema/Acpedidos
            .hbm.xml"/>
        <mapping resource="aida/database/hibernate/sil/schema/Servico.
            hbm.xml"/>
        <mapping resource="aida/database/hibernate/sil/schema/Pedidos.
            hbm.xml"/>
        <mapping resource="aida/database/hibernate/sil/schema/
            Servpostped.hbm.xml"/>
        <mapping resource="aida/database/hibernate/sil/schema/
            PdListaPedidos.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

Listing A.2: Hibernate configuration file for the Pcedoentes database object.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
    3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="aida.database.hibernate.pce.schema.Pcedoentes" table="
        PCEDOENTES">
        <id name="numSequencial" type="long">
            <column name="NUM_SEQUENCIAL" precision="10" scale="0" />
            <generator class="assigned" />
        </id>
        <property name="numProcesso" type="java.lang.Integer">
            <column name="NUM_PROCESSO" precision="8" scale="0" />
        </property>
        <property name="dtaNascimento" type="date">
            <column name="DTA_NASCIMENTO" length="7" />
        </property>
        <property name="nome" type="string">
            <column name="NOME" length="100" />
        </property>
        <property name="sexo" type="string">
```

```
        <column name="SEXO" length="1" />
      </property>
      <property name="morMorada" type="string">
        <column name="MOR_MORADA" length="50" />
      </property>
      <property name="locMorada" type="string">
        <column name="LOC_MORADA" length="20" />
      </property>
      <property name="codPostal" type="java.lang.Short">
        <column name="COD_POSTAL" precision="4" scale="0" />
      </property>
      <property name="codIndicativo" type="string">
        <column name="COD_INDICATIVO" length="3" />
      </property>
      <property name="telMorada" type="java.lang.Long">
        <column name="TEL_MORADA" precision="10" scale="0" />
      </property>
      <property name="dtaRegisto" type="date">
        <column name="DTA_REGISTO" length="7" />
      </property>
      <property name="numMecanografico" type="java.lang.Integer">
        <column name="NUM_MECANOGRAFICO" precision="5" scale="0" />
      </property>
      <property name="desPostal" type="string">
        <column name="DES_POSTAL" length="50" />
      </property>
      <property name="estadoCivil" type="string">
        <column name="ESTADO_CIVIL" length="10" />
      </property>
      <property name="codProfissao" type="java.lang.Long">
        <column name="COD_PROFISSAO" precision="10" scale="0" />
      </property>
      <property name="codHabilitacao" type="java.lang.Long">
        <column name="COD_HABILITACAO" precision="10" scale="0" />
      </property>
    </class>
</hibernate-mapping>
```

Listing A.3: Hibernate POJO configuration file for the Pcedoentes database object.

```
package aida.database.hibernate.pce.schema;

import java.sql.Blob;
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * Pcedoentes generated by hbm2java
 */
@Entity
@Table(name="PCEDOENTES")
public class Pcedoentes implements java.io.Serializable {


    private Long numSequencial;
    private Integer numProcesso;
    private Date dtaNascimento;
    private String nome;
    private String sexo;
    private String morMorada;
    private String locMorada;
    private Short codPostal;
    private String codIndicativo;
    private Long telMorada;
    private Date dtaRegisto;
    private Integer numMecanografico;
    private String desPostal;
    private String estadoCivil;
    private Long codProfissao;
    private Long codHabilitacao;
```

```
public Pcedoentes() {
}



public Pcedoentes(long numSequencial) {
   this.numSequencial = numSequencial;
}
public Pcedoentes(long numSequencial, Integer numProcesso, Date
    dtaNascimento, String nome, String sexo, String morMorada, String
    locMorada, Short codPostal, String codIndicativo, Long telMorada,
    Date dtaRegisto, Integer numMecanografico, String desPostal, String
    estadoCivil, Long codProfissao, Long codHabilitacao) {
  this.numSequencial = numSequencial;
  this.numProcesso = numProcesso;
  this.dtaNascimento = dtaNascimento;
  this.nome = nome;
  this.sexo = sexo;
  this.morMorada = morMorada;
  this.locMorada = locMorada;
  this.codPostal = codPostal;
  this.codIndicativo = codIndicativo;
  this.telMorada = telMorada;
  this.dtaRegisto = dtaRegisto;
  this.numMecanografico = numMecanografico;
  this.desPostal = desPostal;
  this.estadoCivil = estadoCivil;
  this.codProfissao = codProfissao;
  this.codHabilitacao = codHabilitacao;
}

@Id
@Column(name="NUM_SEQUENCIAL", unique=true, nullable=false, precision
    =10, scale=0)
public long getNumSequencial() {
   return this.numSequencial;
}
```

```
public void setNumSequencial(long numSequencial) {
   this.numSequencial = numSequencial;
}

@Column(name="NUM_PROCESSO", precision=8, scale=0)
public Integer getNumProcesso() {
   return this.numProcesso;
}

public void setNumProcesso(Integer numProcesso) {
   this.numProcesso = numProcesso;
}
@Temporal(TemporalType.DATE)
@Column(name="DTA_NASCIMENTO", length=7)
public Date getDtaNascimento() {
   return this.dtaNascimento;
}

public void setDtaNascimento(Date dtaNascimento) {
   this.dtaNascimento = dtaNascimento;
}

@Column(name="NOME", length=100)
public String getNome() {
   return this.nome;
}

public void setNome(String nome) {
   this.nome = nome;
}

@Column(name="SEXO", length=1)
public String getSexo() {
   return this.sexo;
}
```

```
public void setSexo(String sexo) {
   this.sexo = sexo;
}

@Column(name="MOR_MORADA", length=50)
public String getMorMorada() {
   return this.morMorada;
}

public void setMorMorada(String morMorada) {
   this.morMorada = morMorada;
}

@Column(name="LOC_MORADA", length=20)
public String getLocMorada() {
   return this.locMorada;
}

public void setLocMorada(String locMorada) {
   this.locMorada = locMorada;
}

@Column(name="COD_POSTAL", precision=4, scale=0)
public Short getCodPostal() {
   return this.codPostal;
}

public void setCodPostal(Short codPostal) {
   this.codPostal = codPostal;
}

@Column(name="COD_INDICATIVO", length=3)
public String getCodIndicativo() {
   return this.codIndicativo;
}

public void setCodIndicativo(String codIndicativo) {
```

```
        this.codIndicativo = codIndicativo;
    }

    @Column(name="TEL_MORADA", precision=10, scale=0)
    public Long getTelMorada() {
        return this.telMorada;
    }

    public void setTelMorada(Long telMorada) {
        this.telMorada = telMorada;
    }
    @Temporal(TemporalType.DATE)
    @Column(name="DTA_REGISTO", length=7)
    public Date getDtaRegisto() {
        return this.dtaRegisto;
    }

    public void setDtaRegisto(Date dtaRegisto) {
        this.dtaRegisto = dtaRegisto;
    }

    @Column(name="NUM_MECANOGRAFICO", precision=5, scale=0)
    public Integer getNumMecanografico() {
        return this.numMecanografico;
    }

    public void setNumMecanografico(Integer numMecanografico) {
        this.numMecanografico = numMecanografico;
    }

    @Column(name="DES_POSTAL", length=50)
    public String getDesPostal() {
        return this.desPostal;
    }

    public void setDesPostal(String desPostal) {
        this.desPostal = desPostal;
```

```
    }

    @Column(name="ESTADO_CIVIL", length=10)
    public String getEstadoCivil() {
        return this.estadoCivil;
    }

    public void setEstadoCivil(String estadoCivil) {
        this.estadoCivil = estadoCivil;
    }

    @Column(name="COD_PROFISSAO", precision=10, scale=0)
    public Long getCodProfissao() {
        return this.codProfissao;
    }

    public void setCodProfissao(Long codProfissao) {
        this.codProfissao = codProfissao;
    }

    @Column(name="COD_HABILITACAO", precision=10, scale=0)
    public Long getCodHabilitacao() {
        return this.codHabilitacao;
    }

    public void setCodHabilitacao(Long codHabilitacao) {
        this.codHabilitacao = codHabilitacao;
    }

}
```

Listing A.4: Hibernate table structure XML file for the persistent log database table.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
   3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
```

```
<class name="aida.database.hibernate.madip.schema.MadipAgentLogs" table=
   "MADIP_AGENT_LOGS">
   <composite-id name="id" class="aida.database.hibernate.madip.schema.
      MadipAgentLogsId">
      <key-property name="platformId" type="string">
         <column name="PLATFORM_ID" length="100" />
      </key-property>
      <key-property name="agentId" type="string">
         <column name="AGENT_ID" length="50" />
      </key-property>
      <key-property name="logDate" type="date">
         <column name="LOG_DATE" length="7" />
      </key-property>
      <key-property name="logType" type="string">
         <column name="LOG_TYPE" length="20" />
      </key-property>
      <key-property name="logMsg" type="string">
         <column name="LOG_MSG" length="4000" />
      </key-property>
   </composite-id>
   <property name="logReference" type="string">
      <column name="LOG_REFERENCE" length="2000" />
   </property>
</class>
</hibernate-mapping>
```

Listing A.5: HL7 encoded request message processed by the agent.

```
<?xml version="1.0"?>
<O19>
      <Mensaje>
            <cod>199132</cod>
            <tipo>OMG</tipo>
            <evento>O19</evento>
            <apporig>hcis</apporig>
            <appdest>AIDA</appdest>
            <centro>CICA</centro>
            <fechaevn>
```

```
            <dia>0000-00-00</dia>
        </fechaevn>
    </Mensaje>
    <Paciente>
        <cod>******</cod>
        <nhc>*******</nhc>
        <iup>*******</iup>
        <nombre>******</nombre>
        <apell1>****</apell1>
        <apell2>******</apell2>
        <fechanac>1953-11-23</fechanac>
        <sexo>F</sexo>
        <direccion>
            <domicilio>RUA SERPA PINTO N&#xBA; 431 PORTA HAB 55</
              domicilio>
            <codpost>4250</codpost>
            <poblacion>
                <cod>131213124</cod>
                <descr>Cedofeita</descr>
            </poblacion>
            <provincia>
                <cod>1312</cod>
                <descr>Porto</descr>
            </provincia>
            <autonomia>
                <cod>13</cod>
                <descr>Porto</descr>
            </autonomia>
            <pais>
                <cod>620</cod>
                <descr>PORTUGAL</descr>
            </pais>
        </direccion>
        <tarjeta>166565151</tarjeta>
        <dni>165550422</dni>
        <tipodoc>NIF</tipodoc>
        <numerodoc>*********</numerodoc>
```

```
        <lugarnac>
            <poblacion>
                <cod>1091097</cod>
                <descr>Fi&#xE3;es</descr>
            </poblacion>
            <provincia>
                <cod>109</cod>
                <descr>Santa Maria da Feira</descr>
            </provincia>
            <autonomia>
                <cod>1</cod>
            </autonomia>
            <pais>
                <cod>620</cod>
                <descr>PORTUGAL</descr>
            </pais>
        </lugarnac>
        <estado>VIVO</estado>
    </Paciente>
    <Episodio>
        <cod>22637</cod>
        <tipo>DOMI</tipo>
        <cama>
            <unenf>UEPISO2</unenf>
        </cama>
        <medico>
            <cod>20110</cod>
            <codcolegiado>*****</codcolegiado>
            <nombre>Rosa</nombre>
            <apell1>Zulmira</apell1>
        </medico>
        <servicio>BLO_GIN</servicio>
        <centro>CICA</centro>
        <procedencia>
            <tipo>5</tipo>
            <origen>9</origen>
        </procedencia>
```

```
<medicoingr>
    <cod>20110</cod>
    <codcolegiado>*****</codcolegiado>
    <nombre>Rosa</nombre>
    <apell1>Zulmira</apell1>
</medicoingr>
<garante>
    <cod>935640</cod>
    <descr>ADSE - SNS</descr>
    <tipo>1</tipo>
</garante>
<motivoingr>
    <cod>1</cod>
    <descr>Ordem M&#xE9;dica</descr>
</motivoingr>
<fechaini>
    <dia>2012-07-09</dia>
    <hora>08:34:00</hora>
</fechaini>
<intervencion>
    <preoperatorio>U</preoperatorio>
    <anestesia>
        <tipo>N</tipo>
        <duracion/>
    </anestesia>
</intervencion>
<diagnostico>
    <cod>20002535</cod>
    <codext>2189</codext>
    <descr>LEIOMIOMA UTERINO, NAO ESPECIFICADO</descr>
</diagnostico>
<procedimiento>
    <cod>20030447</cod>
    <codext>6829</codext>
    <descr>EXCISAO OU DESTRUICAO DE LESAO DO UTERO, NCOP</
      descr>
</procedimiento>
```

```
</Episodio>
<Pedido>
    <cod>7724</cod>
    <estado>EMITIDO</estado>
    <servicio>HGSAAPA</servicio>
    <tipo>DOMI</tipo>
    <prioridad>
        <cod>1</cod>
        <descr>Normal</descr>
    </prioridad>
    <procedencia>
        <servicio>BLO_GIN</servicio>
    </procedencia>
    <medicopeti>
        <cod>20110</cod>
        <codcolegiado>*****</codcolegiado>
        <nombre>Rosa</nombre>
        <apell1>Zulmira</apell1>
    </medicopeti>
    <operador>
        <cod>3672</cod>
    </operador>
    <observaciones>mioma uterino sub mucoso assintomatico</
      observaciones>
    <diagnostico>
        <cod>20002535</cod>
        <codext>2189</codext>
        <descr>LEIOMIOMA UTERINO, NAO ESPECIFICADO</descr>
    </diagnostico>
    <garante>
        <cod>935640</cod>
        <descr>ADSE - SNS</descr>
        <tipo>1</tipo>
    </garante>
    <peticion>
        <cod>5695</cod>
        <accion>nuevo</accion>
```

```
            <estado>EMITIDA</estado>
            <fechapet>
                  <dia>2012-07-09</dia>
                  <hora>12:43</hora>
            </fechapet>
            <prestacion>
                  <cod>2040</cod>
                  <codext>B31016</codext>
                  <descr>Histologia (ANAPAT)</descr>
            </prestacion>
            <muestra>
                  <cod>4233</cod>
                  <volumen>
                        <valor>1.0</valor>
                        <unidades>Outra</unidades>
                  </volumen>
            </muestra>
      </peticion>
   </Pedido>
   <PatientPT>
      <seqNumberId>*****</seqNumberId>
      <fatherName>***** ***** *********</fatherName>
      <motherName>**** ****** *****</motherName>
      <maritalStatus>33</maritalStatus>
      <codPost2>466</codPost2>
      <sonhoPostdescr>Porto</sonhoPostdescr>
      <telefmovil>**********</telefmovil>
      <nacionalidad>Portuguesa</nacionalidad>
      <grusubsist>1</grusubsist>
      <subsist>935640</subsist>
      <num_beneficiario>***********</num_beneficiario>
      <creationdateext>2011-08-30</creationdateext>
   </PatientPT>
</O19>
```